**BEZIEHUNGSTYPEN**: **Einfache Beziehung** 1:Jede Entität (Zeile) der ersten Entitätsmenge (Tabelle) steht mit genau einer Entität (genau 1) der zweiten Entitätsmenge in Beziehung. **Konditionelle Beziehung** c:Jede Entität der ersten Entitätsmenge kann mit höchstens einer Entität (0 oder 1) der zweiten Entitätsmenge in Beziehung stehen. **Mehrfach Beziehung** m: Jede Entität der ersten Entitätsmenge steht mit mindestens einer Entität (1 oder mehrere) der zweiten Entitätsmenge in Beziehung. **Mehrfachkonditionelle Beziehung** mc: Jede Entität der ersten Entitätsmenge kann mit beliebig vielen Entitäten (0, 1 oder mehrere) der zweiten Entitätsmenge in Beziehung stehen.

Es gibt drei Arten von Beziehungen:

* Eine 1:1-Beziehung verknüpft einen Datensatz einer Tabelle mit genau einem Datensatz einer anderen Tabelle. Dies ist immer dann nützlich, wenn bestimmte Informationsaspekte nicht so oft benötigt werden wie andere Aspekte. Diese Beziehung kommt in der Praxis am wenigsten oft vor.
* Eine 1:m-Beziehung oder Eins-zu-viele-Beziehung verbindet einen Datensatz einer Tabelle mit beliebig vielen Datensätzen einer anderen Tabelle. Dies ist der häufigste Verknüpfungstyp. Detailinformationen über Werte, die in einer Spalte einer Tabelle oft vorkommen können, werden in einer separaten Tabelle erfasst.  
  Beispiel: Oben sehen wir eine 1:mc-Beziehung. In jedem Ort wohnen 0, 1 oder mehrere Lernende, jeder Lernende wohnt an genau 1 Ort.
* Eine m:m-Beziehung, auch Viele-zu-viele-Beziehung genannt, kombiniert beliebig viele Vorkommen eines bestimmten Wertes mit beliebig vielen Vorkommen eines anderen.  
  Beispiel: Jeder Lernende besucht mehrere Module, jedes Modul wird von mehreren Lernenden besucht

Um den **Beziehungstyp zu bestimmen**, geht man wie folgt vor:

1. **Erstens** Als Ausgangslage braucht es zwei Entitätsmengen und eine Beschreibung der Beziehung (Verb, Substantiv oder kurzer Satz). Beispiel: Entitätsmenge 1 = Kunde, Entitätsmenge 2 = Ort, Beziehung = wohnt bzw. wohnen
2. **Zweitens** Um die Kardinalität auf der rechten Seite (Ort) zu bestimmen, geht man von der Menge aller Kunden aus: Jeder Kunde wohnt an genau einem Ort **->** Kardinalität = 1.
3. **Drittens** Um die Kardinalität auf der linken Seite (Kunde) zu bestimmen, geht man von der Menge aller Orte aus: An jedem Ort wohnen 0, 1 oder mehrere Kunden **->** Kardinalität = mc.
4. **Demnach** ist die Beziehung mc:1

!!! In Relationalen Datenbanken ist nur 1:1,1:m und m:mc möglich. Um m:m Beziehungen darzustellen benötigt man eine Zwischentabelle Eine m:m-Beziehung müssen wir mit einer Zwischentabelle abbilden. Wenn in einer Bestellung 10 Filme bestellt werden, dann gibt es in der Zwischentabelle 10 Eintragungen. Denn wir wollen nicht in der Tabelle *Bestellungen* zehn Spalten mit Fremdschlüsseln haben. Das wäre unflexibel (was ist, wenn jemand 20 Filme bestellt?) und würde die Auswertungen wie in Excel erschweren.Wenn ein Film von 12 verschiedenen Personen bestellt wird, dann gibt es 12 Bestellungen mit unterschiedlichem Datum und Betrag. Es wäre wiederum unflexibel und unpraktisch, die Bestellungen bei den Filmen festzuhalten (in diesem Fall wären das 12 Spalten mit Fremdschlüsseln).

**UNSTRUKTURIERTE DATEN** können nicht in einer Tabelle organisiert werden, sondern brauchen meist eigene Software für die Organisation (zb. Musik).

* **Semistrukturierte Daten** haben zwar Struktur, diese ist jedoch von Objekt zu Objekt unterschiedlich, wie zb Word Dokumente.
* **Strukturierte Daten** können in DBs gespeichert werden und haben ein vorgegebenes Format.

**BEGRIFFE** Entitätstyp/Tabellendefinition = Der Name einer Tabelle. Entität/Tupel = Eine Zeile. Entitätsmenge/Tabelle = Die Menge aller Datenobjekte mit dem selben Entitätstyp sprich alle Datensätze einer Tabelle. Attribut/Merkmal = Die Spalten. Datentypen = String/Int/etc. Wertebereich/Domäne = Der zugelassene Bereich eines Wertes zb. Postleitzahl muss vierstellig sein. Attributwert = Der genaue Wert eines Attributes einer Entität, also der Wert einer Zelle der Tabelle. Primärtabelle = die Tabelle mit den Primärschlüsselwerten, auf die in der Detailtabelle verwiesen wird. Detailtabelle = Die Tabelle mit dem Fremdschlüsselwerten, die auf die Primärschlüsselwerte der Primärtabelle verweisen. Konsistenz = Die Freiheit von Wiedersprüchen in einer Datenbank, wenn alle Redundanzen beseitigt sind // Konsistenz = Die Übereinstimmung von an verschiedenen Stellen gespeicherte Daten oder des Bezuges zwischen Daten in der DB. Zwischentabelle = eine Tabelle die zum erstellen von DBs mit m:m Beziehung gebraucht wird.

**Konsistenzgarantie durch ACID** steht für **A**tomicity (Atomarität), **C**onsistency (Konsistenz), **I**solation, **D**urability (Dauerhaftigkeit). **ANOMALIEN** sind: **Update-Anomalie**, wenn gleiche Daten unterschiedlich benennt werden und dadurch bei Updates nur Teile der DB verändert werden. **Delete-Anomalie**, wenn beim Löschen einer Entität ungewollt auch andere Informationen verloren gehen. **Insert-Anomalie**, wenn Attribute fehlen, mehrere Primärschlüssel dieselbe Information tragen oder ein Objekt mehrere Schlüssel trägt.

**ABHÄNGIGKEITEN**Ein Bild, das Text, Screenshot, Schrift, Zahl enthält.

Automatisch generierte Beschreibung

1. **funktionale Abhängigkeit** liegt dann vor, wenn ein Nichtschlüsselattribut nur von einem Teil der Attribute eines zusammengesetzten Schlüsselkandidaten funktional abhängig ist. Beispiel: Das Datum ist funktional abhängig von der Rechnungsnummer *RNr*, nicht aber vom gesamten Schlüssel *RNr* und *PNr*.
2. **vollständig funktionale Abhängigkeit** liegt dann vor, wenn ein Nichtschlüsselattribut nicht nur von einem Teil der Attribute eines zusammengesetzten Schlüsselkandidaten funktional abhängig ist, sondern von allen Teilen des Primärschlüssels. Die **vollständig funktionale Abhängigkeit** wird mit der zweiten Normalform erreicht. Beispiel: Der Artikel ist vollständig funktional abhängig vom gesamten Primärschlüssel *RNr* und *PNr*.
3. **transitive Abhängigkeit** liegt dann vor, wenn ein Nichtschlüsselattribut funktional abhängig ist von einem andern Nichtschlüsselattribut. Es ist transitiv oder indirekt abhängig vom Primärschlüssel.Beispiel: Name und Adresse sind abhängig von der Kundennummer *KNr*. Somit sind diese Attribute transitiv abhängig vom Schlüsselkandidaten.

**NORMALFORMEN**- **Definition Nullte Normalform:** Eine Tabelle befindet sich in der **nullten Normalform**, wenn alle Datenelemente der realen Welt darin zusammengefasst und aufgelistet sind. **erste Normalform:** wenn alle Daten **atomar vorliegen**und alle Tabellenspalten gleichartige Werte enthalten **Definition zweite Normalform:** wenn sie sich in der ersten Normalform befindet und zudem jedes Nichtschlüsselattribut vom Primärschlüssel **voll funktional abhängig** ist. **Definition dritte Normalform:** wenn er sich in der zweiten Normalform befindet und es zudem ***keine transitiven Abhängigkeiten*** gibt.

* **Normalisierung** ist die Überführung einer Datenbanktabelle in eine Normalform höheren Grades. Es ist die Aufteilung von Attributen in mehrer Tabellen mit dem Ziel, Redundanzen zu verringern (bzw komplett zu verhindern)
* **Nullte Normalform**: Eine Tabelle ist in der nullten Normalform, wenn alle Daten der realen Welt darin unstrukturiert aufgefasst sind. In so einer Tabelle können in einem Attributwert auch mehrere Informationen gelistet sein (was in einer DB nicht vorkommen sollte (atomarität)).
* **Erste Normalform**: Wenn alle Daten atomar vorliegen (pro Attributwert nur ein Wert), ist die Tabelle in der ersten Normalform. Man trennt also alle Informationen in die kleinst mögliche Einheit auf (die Sinn ergibt) und gibt ihr eine eigene Spalte (macht sie zu einem eigenen Attribut)
* **Zweite Normalform**: Wenn die Attribute einer Tabelle nur von Primärschlüsseln abhängig sind, ist sie automatisch bereits in der zweiten Normalform. Der Unterschied zur ersten Normalform besteht darin, dass jedes Attribut von den Schlüsseln abhängig ist. Es können jedoch immernoch Abängigkeiten von Attributen untereinander bestehen wie zb. die Abhängigkeit von Name zu Kundennummer.
* **Dritte Normalform**: In Tabellen der dritten Normalform sind alle Attribute der Tabelle nur noch von ihrem Primärschlüssel und nicht mehr von anderen Attributen abhängig. Es werden sperate Tabellen gemacht, die nur dafür da sind, die Schlüssel verschiedener Tabellen (Einkäufe, Kunden) zu verbinden.