1. Grundlagen Datenmodellierung

Datenmodelle

Um in das Thema zu starten, stelle ich die Frage, was ein Datenmodell bzw. Datenmodellierung überhaupt ist? Wikipedia hat dazu natürlich auch eine Antwort:

„In der Informatik, im Besonderen bei der Entwicklung von Informationssystemen, dienen ***Datenmodelle*** und die zu deren Erstellung durchgeführten Aktivitäten (***Datenmodellierung***) dazu, die Struktur für die in den Systemen zu verarbeitenden (im Besonderen für die zu speichernden) Daten zu finden und festzulegen.“ (Wikipedia -> Datenmodell)

Es geht also darum, Informationsstrukturen zu modellieren, um dann darauf basierend die Implementierung von Datenbanken umzusetzen. Datenmodellierung ist also eine Phase innerhalb der Datenbankentwicklung.

In der Phase Datenmodellierung werden dabei drei Datenmodelle entwickelt, die aufeinander aufbauen:

1. Konzeptionelles Datenmodell
2. Logisches Datenmodell
3. Physisches Datenmodell

Das ***konzeptionelle Datenmodell*** ist dabei implementierungsunabhängig und bildet somit reine Fachlichkeit ab. Es wird direkt aus den Anforderungen heraus entwickelt. Das ***logische Datenmodell*** ist die Abbildung des konzeptionellen Datenmodells auf das zu verwendende Datenbanksystem, d.h. ein relationales Datenmodell, etc. Das ***physische Datenmodell*** erweitert dann das logische Datenmodell um technische Aspekte (Indizes, Partitionierung, etc.)

Es gibt noch unterschiedliche Arten der Modellierung, die für unterschiedliche Zwecke zum Einsatz kommen:

- 3. Normalform Modellierung (3 NF Modell)
- Star Schema
- Data Vault
- Galaxy Schema
- …

Die ***3 NF Modellierung*** findet man hauptsächlich in operativen Systemen, d.h. Systeme in denen die Dateneingaben an sich stattfinden (z.B. ERP Systeme). Darüber hinaus auch im Core DWH oder Enterprise DWH. Dabei sollen die Daten in anwendungsfreier Form gespeichert werden. Das ***Star Schema*** stellt eine besondere Art der Modellierung dar, die vor allem auf Auswertungssysteme ausgelegt ist. Diese findet man dann eben auch größtenteils im Reporting Layer (Data Mart) eines Datawarehouse Systems. Die ***Data Vault Modellierung*** ist eine recht neue Form der Modellierung. Diese zielt auf eine einfache Erweiterbarkeit und möglichst automatisierte Erstellung der Beladejobs ab. Man findet diese Form der Modellierung dann auch hauptsächlich im Bereich des Core DWH, Integration Layer, etc.

Anforderungen und Ihre Aufnahme

Um ein Datenmodell zu entwickeln sind ***Anforderungen*** notwendig. Anforderungen ergeben sich aus den Wünschen der zukünftigen Anwender des Systems. Wie in allen Softwareprojekten so kommt auch den Anforderungen in Datenbankprojekten eine wesentliche Bedeutung zu. Viele Projekte scheitern bzw. verzögern sich deutlich aufgrund fehlender oder qualitativ schlechter Anforderungen.

Anforderungen dienen dabei als Grundlage für verschiedene Zwecke:

- Kommunikation
- Ausschreibung und Vertragsgestaltung
- Systemintegration, Wartung und Pflege
- Systemarchitektur

Anforderungen dienen allen Beteiligten als Kommunikations-, Diskussions- und Argumentationsgrundlage. Weiterhin dienen Anforderungen als Grundlage für Ausschreibungen und Vertragsgestaltung, da die Anforderungen den Umfang des zu entwickelnden Systems beschreiben. Fehlen Anforderungen komplett, kann keine seriöse Aufwandsschätzung stattfinden und damit fehlen wesentliche Aspekte einer Vertragsgestaltung. Da Anforderungen die Wünsche der Anwender beschreiben, haben diese natürlich auch Auswirkung auf die Systemarchitektur. Falsche Anforderungen zu Datenmengen und Abfragezeiten können zu komplett unterschiedlichen Architekturen führen und im Ergebnis zu schlechter Performance.

Aufgrund der oben dargestellten Funktionen der Anforderungen und damit einhergehenden Einfluss auf die erfolgreiche Implementierung, sollte der Phase Anforderungsaufnahme eine hohe Priorität beigemessen werden. Fehler während der Anforderungsaufnahme auszumerzen ist nicht teuer. Fehler, die erst im Test auffallen sind um ein vielfaches teurer, weil sämtliche Phasen im Entwicklungsprozess erneut durchlaufen werden müssen.

Wie erhält man nun qualitativ hochwertige Anforderungen? Indem man jede Anforderung hinterfragt. Fragen und Hinterfragen ist das A und O bei der Ermittlung und Klärung von Anforderungen. In der Praxis treten bei der Kommunikation diverse Effekte auf, wodurch für den Empfänger wesentliche Informationen fehlen, verallgemeinert oder verzerrt wurden. Im Folgenden sollen nur die 6 häufigsten Effekte einmal dargestellt werden:

1) Nominalisierungen (Verb → Substantiv)
2) Unvollständig spezifizierte Prozesswörter (W-Fragen?)
3) Substantive ohne Bezugsindex (die Daten, die Anwender → Welche genau?)
4) Unvollständig spezifizierte Bedingungen
5) Modaloperatoren der Notwendigkeit
6) Implizite Annahmen

Darüber hinaus gibt es noch viele weitere Effekte, wodurch bei der Kommunikation viele Informationen verloren gehen. Nähere Informationen dazu findet man z.B. im Buch „Requirements Engineering“ von Chris Rupp et al.

Betrachten wir einmal den Punkt 2 – unvollständig spezifizierte Prozesswörter. Bestimmte Verben erfordern mehr als ein Hauptwort. Das Verb „übertragen“ erfordert zumindest drei weitere Informationen: Was, woher und wohin.

Man kann jetzt bestimmte Regeln anwenden, um solche Unvollständigkeiten von vornerein zu vermeiden:

1) Formulieren Sie jede Anforderung im Aktiv. → Dabei muss immer ein Subjekt angeben.
2) Drücken Sie Prozesse durch Vollverben aus → Dadurch wird deutliche, welche Bestandteile bei dem Verb fehlen.
3) Decken Sie unvollständig spezifizierte Prozesswörter auf → Dadurch werden weggelassene Satzbestandteile spezifiziert.

Für alle anderen Effekte gibt es ebenfalls diverse Regeln, wie man diese finden bzw. gar nicht erst auftreten lassen kann. Dieses Regelwerk findet sich ebenfalls in dem Buch „Requirements Engineering“ von Chris Rupp et al.

Darüber hinaus gibt es eine Schablone, um Anforderungen sauber zu definieren. Diese sieht wie folgt aus:

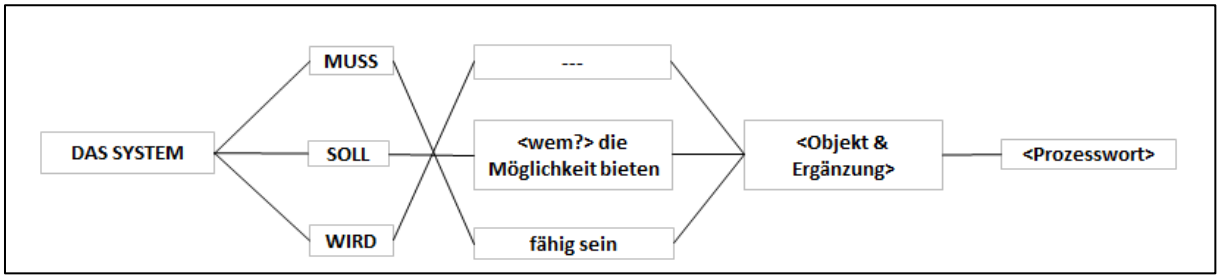

Abb.: Anforderungsschablone ohne Bedingung

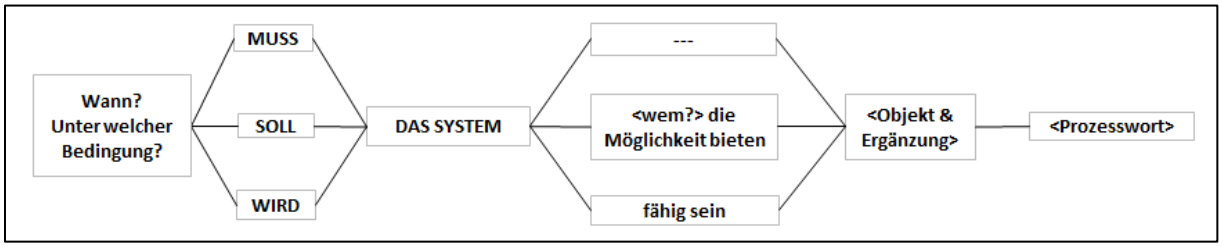

Abb.: Anforderungsschablone mit Bedingung

Ein paar Beispiele dazu:

1) Selbstständige Systemaktivität:

   Wenn ein Nutzer innerhalb eines Ausleihvorgangs einen Bibliothekskunden auswählt, soll das Bibliothekssystem diesem Nutzer den Namen des Kunden, die Adresse des Kunden und den augenblicklichen Kontostand des Kunden anzeigen.

2) Benutzerinteraktion:

   Falls ein Bibliothekskunde weniger als 10 Leihobjekte entliehen hat, soll das Bibliothekssystem einem Nutzer die Möglichkeit bieten, Leihobjekte online zu reservieren.

3) Schnittstellenanforderung:

   Wenn das Bibliothekssystem in Betrieb ist, soll das System fähig sein, Daten für ein Software Update von einem zentralen Admin-Server über das lokale Netzwerk zu empfangen.

Wenn man zur Anforderungsdefinition diese Schablonen verwendet, bekommt man automatisch schon bessere Anforderungen. Wenn man darüber hinaus die oben beispielhaft vorgestellten Regeln auf das Ergebnis anwendet, bekommt man qualitativ hochwertige Anforderungen.

Eine andere Möglichkeit ist, anhand der folgenden Schritte vorgegebene Anforderungen zu überprüfen und ggf. weitere Fragen zu stellen, um wirklich alle Aspekte und Informationen zu bekommen:

1) **Prozesswörter** überprüfen → Verben und substantivierte Verben (z.B. melden, Auslastung, erfassen). Stellen Sie zu jedem dieser Prozesswörter die W-Fragen: Wer? Was? Wann? Wie? Wo? Warum?
2) **Komparative & Superlative** überprüfen → Worauf bezieht sich der Vergleich oder die Steigerung?
3) **Universalquantoren** (alle, keiner, immer, nie, jeder, stets, …) überprüfen → Hinterfragen Sie Ausnahmen & implizite Annahmen
4) **Bedingungen** überprüfen → Vollständig? Alle Entscheidungs-/Verzweigungsbedingungen genannt?
5) **Konstanten und konfigurierbare Werte** überprüfen → Identifikation und Vergabe sprechender Namen
6) **Abkürzungen & Fachbegriffe im Glossar definieren**

<u>Beispiele:</u>

1) Das System muss performanter sein.
    - ➔ Performanter im Vergleich wozu?
    - ➔ Was genau muss performant sein? Antwortzeit? Zeit für Laderoutinen?

2) Keiner darf Veränderungen an den Einstellungen vornehmen
    - ➔ Keiner Ihrer Mitarbeiter?
    - ➔ Wer stellt die Werte dann initial ein?

Aufgaben

1) Lösen Sie das ausgeteilte Kreuzworträtsel.

2) Überprüfen Sie die folgenden Anforderungen auf Qualität. Was genau ist an den Anforderungen nicht so gut? Welche Fragen sollte man als Entwickler stellen?
    a. Die Daten sollen immer gesichert werden.
    b. Keiner darf das Berechnungsmodul ohne entsprechendes Passwort starten.
    c. Alle Berichte sollen schnell sein.
    d. Die Durchführung von Berechnungen muss sichergestellt sein.
    e. Das System darf nie ausfallen.

2. Entity Relationship Modell

Grundlagen des ER Modells

Das ER Modell soll eine Abbildung der realen Welt darstellen. Es geht darum für bestimmte Problembereiche und Aufgabenstellungen eine vereinfachte Darstellung des realen Problems zu erreichen. Dabei werden nur für die Aufgabenstellung relevante Aspekte abgebildet. Das ER Modell geht zurück auf Peter Sin-Shan Chen und wurde im Jahre 1976 erstmals veröffentlicht.

Es werden Objekte (Entity) aus der realen Welt abgebildet und deren Beziehungen (Relationships) untereinander. Eine ***Entität*** ist dabei ein individuelles, unterscheidbares und identifizierbares Exemplar von Dingen, Personen oder Begriffen der realen oder der Vorstellungswelt.

Eine Entität kann in der Praxis alles Mögliche sein, z.B. ein Individuum(Kunde, Mitarbeiter, Lieferant), ein realer Gegenstand (Rechnung, Auto, Produkt,…) oder auch ein abstraktes Konzept (Produktplan, Sprache, …)

Entitäten beschreiben also Objekte und werden näher definiert durch ihre ***Attribute***. Attribute (oder auch Eigenschaften) beschreiben bestimmte Merkmale der Entität. Wenn man die Entität Auto betrachtet, hat diese dann eine Farbe, einen Hersteller, ein Modell, eine Höchstgeschwindigkeit, etc. Dieses sind alles Beispiele für Eigenschaften der Entität Auto.

Eine Entität ist die Beschreibung von gleichen Objekten. In der Praxis kann es dann mehr als ein konkretes Exemplar einer Entität geben. Beispiel: Entität ist das Auto. Die Exemplare sind dann ein VW Golf, ein Audi A3, BMW 5er, ….

Um die einzelnen Exemplare einer Entität eindeutig identifizieren und unterscheiden zu können gibt es sog. ***Entitätsschlüssel***. Das sind dann eine oder mehrere Eigenschaften der Entität, die zusammengenommen eine eindeutige Identifizierbarkeit zulassen.

Manchmal gibt es keine Eigenschaft oder Kombination von Eigenschaften, mit deren Hilfe sich jedes konkrete Exemplar einer Entität identifizieren lässt. Dann benötigt man einen ***künstlichen Schlüssel***, d.h. man schafft ein neues Attribut, das nur dem Zwecke der eindeutigen Identifizierbarkeit dient. Dieser muss folgende Anforderungen erfüllen:

- Eindeutigkeit
- Unveränderlichkeit
- Sofortige Zuteilbarkeit

Ein künstlicher Schlüssel wird auch Surrogate Key bezeichnet. Im Gegensatz dazu gibt es noch den natürlichen Schlüssel (der dann aus „normalen“ Attributen der Entität besteht.)

Im Regelfall existieren in einem Modell mehr als eine Entität. Zusammenhänge zwischen zwei und mehr Entitäten werden durch ***Beziehungen*** realisiert. Beziehungen werden durch ein aktives oder passives Verb benannt. Eine Beziehung verbindet zwei oder mehr Entitäten wechselseitig. Beziehungen zwischen Entitäten können nur existieren, wenn die Entitäten existieren. Beziehungen werden durch ihre Entitäten identifiziert, bzw. genauer durch ihre Entitätsschlüssel.

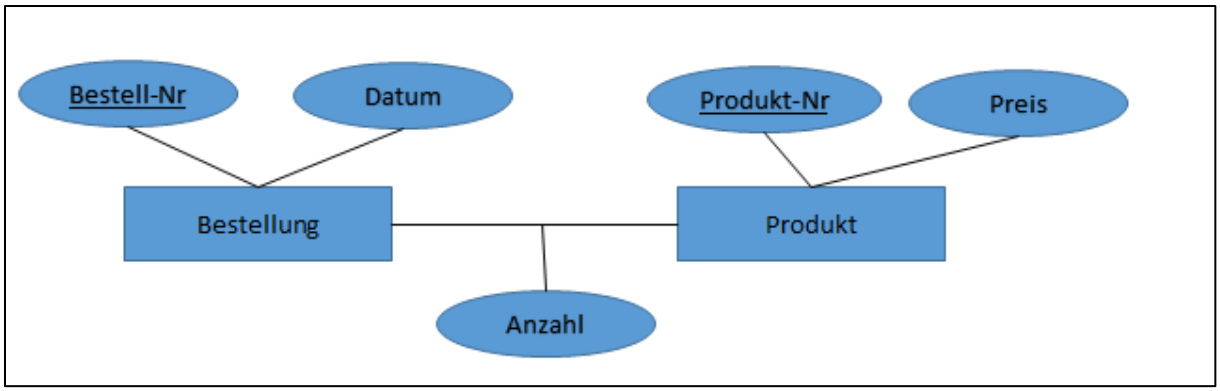

Abb.: Entitäten, Attribute, Primärschlüssel, Beziehung & Beziehungsattribute

Beziehungen können ebenfalls Attribute enthalten, dann spricht man von ***Beziehungseigenschaften***. Diese Eigenschaften beschreiben dann die Beziehung im Detail.

Die ***Stelligkeit*** einer Beziehung beschreibt, wie viele Entitäten durch die Beziehung miteinander verbunden sind.

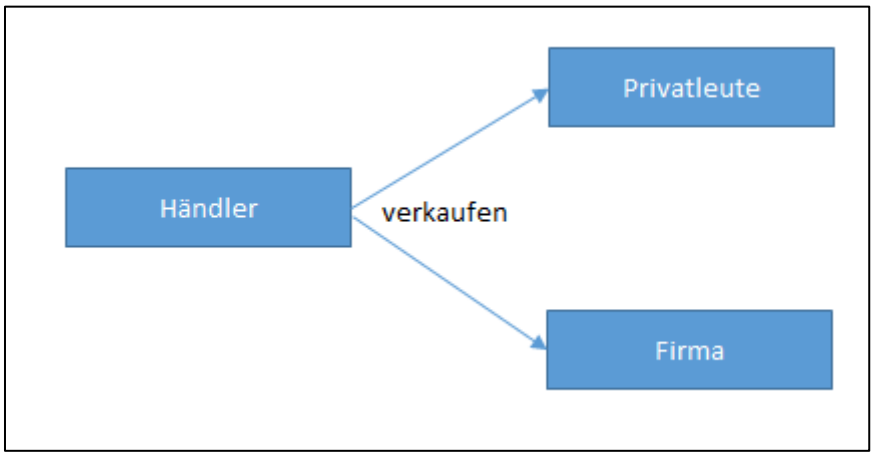

Abb.: Mehrseitige Beziehung

Mehrseitige Beziehungen können oftmals in zweiseitige Beziehungen aufgelöst werden, indem eine neue Entität geschaffen wird und die übrigen Entitäten direkt mit dieser verbunden werden. Dieses ist möglich, da viele Dinge der Realität entweder als Entität oder als Beziehung aufgefasst werden können in Abhängigkeit davon ob man es als Verb oder Substantiv beschreibt. Es gibt zum Beispiel eine Lieferung oder Entität A liefert etwas an Entität B. Verkauf oder verkaufen, etc.

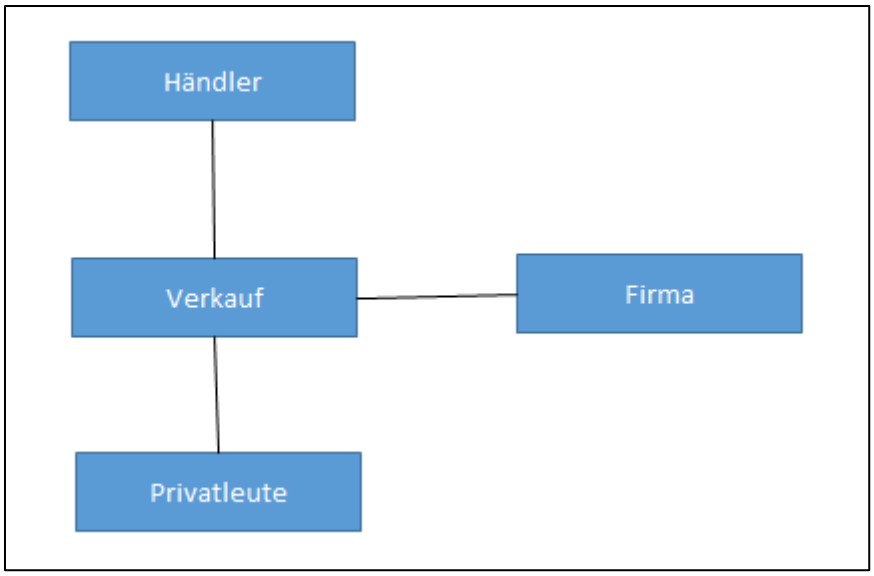

Abb.: Vereinfachte Beziehungen

Die ***Kardinalität*** einer Beziehung gibt an, mit wie vielen Entitäten derselben Art eine betrachtete Entität in Beziehung steht. Eine Kardinalität ist also eine Anzahl und zwar eine der folgenden vier:

- ***Einfache Kardinalität (1)***
  Es steht eine Entität mit genau einer anderen Entität in Beziehung.
- ***Bedingte Kardinalität (C)***
  Es steht eine Entität mit höchstens einer anderen Entität in Beziehung.
- ***Mehrfache Kardinalität (M)***
  Es steht eine Entität mit mindestens einer anderen Entität in Beziehung.
- ***Bedingt mehrfache Kardinalität (MC)***
  Eine Entität kann mit beliebig vielen anderen Entitäten in Beziehung stehen.

Die Kardinalität wird für jede Beziehung für beide Entitäten angegeben, d.h. die Beziehung wird aus beiden Richtungen betrachtet.

Dazu ein Beispiel:

1. Ein BERATER berät mehrere KUNDEn.
2. Ein KUNDE wird von mehreren BERATERn beraten.

Betrachtet man nur 1) für sich alleine, ist nicht klar, ob jeder Kunde exklusiv von einem Berater beraten wird. Erst durch 2) wird klar, dass in diesem Fall jeder Kunde von mehereren Beratern beraten wird. Deshalb handelt es sich hier um eine n:m Beziehung.

Im Ergebnis resultieren daraus die folgenden Beziehungsarten:

1. 1:1 Beziehungen
2. 1:C Beziehungen
3. 1:N Beziehungen
4. 1:NC Beziehungen
5. C:C Beziehungen
6. C:N Beziehungen
7. C:NC Beziehungen
8. N:M Beziehungen
9. N:CM Beziehungen

10. CN:CM Beziehungen

Beispiele: KUNDE <---------> BERATER

Beziehungsart	Anforderungen
1:1	1. Ein KUNDE wird von genau einem BERATER beraten. 2. Ein BERATER berät genau einen KUNDE.
1:C	1. Ein KUNDE wird von einem oder keinem BERATER beraten. 2. Ein BERATER berät genau einen KUNDE.
1:N	1. Ein KUNDE wird von mehreren BERATER beraten. 2. Ein BERATER berät genau einen KUNDEn
1:NC	1. Ein KUNDE wird von mehreren, keinem oder einen BERATER beraten. 2. Ein BERATER berät genau einen KUNDEn
C:C	1. Ein KUNDE wird von einem oder keinem BERATER beraten. 2. Ein BERATER berät einen oder keinen KUNDE.
C:CN	1. Ein KUNDE wird von einem, keinem oder mehreren BERATERn beraten. 2. Ein BERATER berät einen oder keinen KUNDE.
1:CN	1. Ein KUNDE wird von einem, keinem oder mehreren BERATERn beraten. 2. Ein BERATER berät genau einen KUNDE.
N:M	1. Ein KUNDE wird von mehreren BERATERn beraten. 2. Ein BERATER berät mehrere KUNDEn.
N:CM	1. Ein KUNDE wird von einem, keinen oder mehreren BERATERn beraten. 2. Ein BERATER berät mehrere KUNDEn.
CN:CM	1. Ein KUNDE wird von einem, keinen oder mehreren BERATERn beraten. 2. Ein BERATER berät einen, keinen oder mehrere KUNDEn.

Übung 1: Geben Sie Beispiele für jede Beziehungsart aus Ihrer beruflichen Praxis.

In der Praxis gibt es häufig sogenannte Spezialisierungen und Generalisierungen von Entitäten. Es gibt dann z.B. eine Entität BERATER und dazu noch „spezielle“ Berater, das wären z.B. FINANZBERATER und ITBERATER. Das hat den Vorteil, dass die Entität BERATER nicht alle Attribute beinhalten muss, sondern nur die für alle Berater relevanten. Spezielle IT Berater Attribute finden sich dann nur in der Entität ITBERATER. In diesem Beispiel ist die Entität BERATER der sog. ***Supertyp*** und die anderen beiden Entitäten sind sog. ***Subtypen***.

Auch zwischen Supertypen und Subtypen existieren die schon vorgestellten Beziehungen.

In der Praxis existieren viele Entitäten mit diversen Beziehungen und zugehörigen Kardinalitäten. Diese Dinge werden dann in einem sog. ***Entity Relationship Diagramm*** dargestellt. So ein Diagramm kann wie folgt aussehen:

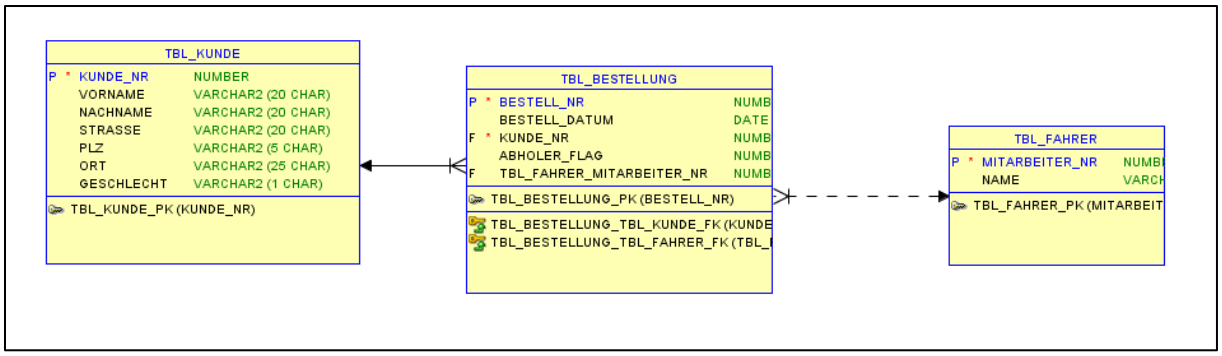

Abb.: Ganz einfaches ER Modell als Beispiel