Datenbanken

Kevin Bertolini | 22.01.2019



Inhaltsverzeichnis

[1. GRUNDLAGEN 1](#_Toc535775561)

[1.1 DB, DBS und DBMS 1](#_Toc535775562)

[1.2 Aufgaben eines Datenbanksystems 2](#_Toc535775563)

[1.3 Entity-Relationship Modell (ERM) 3](#_Toc535775564)

[1.4 Entity-Relationship Diagramm (ERD) 3](#_Toc535775565)

[1.5 Relationale Datenbank 4](#_Toc535775566)

[1.6 Hierarchische Datenbank 4](#_Toc535775567)

[1.7 Persönliches Fazit zu Kapitel 1 5](#_Toc535775568)

[2. DATENBANKSPRACHE 6](#_Toc535775569)

[2.1 Structured Query Language (SQL) 6](#_Toc535775570)

[2.2 Data Control Language (DCL) 6](#_Toc535775571)

[2.3 Data Definition Language (DDL) 7](#_Toc535775572)

[2.4 Data Query Language (DQL) 7](#_Toc535775573)

[2.5 Persönliches Fazit zu Kapitel 2 8](#_Toc535775574)

[3. BEGRIFFE EINER DATENBANK 9](#_Toc535775575)

[3.1 Entität 9](#_Toc535775576)

[3.2 Entitätsmenge 9](#_Toc535775577)

[3.3 Tabelle 9](#_Toc535775578)

[3.4 Datensatz 10](#_Toc535775579)

[3.5 Attribut 11](#_Toc535775580)

[3.6 Feld 11](#_Toc535775581)

[3.7 Wert /Wertbereich 12](#_Toc535775582)

[3.8 Der Begriff Null 12](#_Toc535775583)

[3.9 Persönliches Fazit zu Kapitel 3 12](#_Toc535775584)

[4. RELATION & SCHLÜSSEL 13](#_Toc535775585)

[4.1 Beziehung (Relation) 13](#_Toc535775586)

[4.2 Primärschlüssel 13](#_Toc535775587)

[4.3 Fremdschlüssel 13](#_Toc535775588)

[4.4 Normalisierung 14](#_Toc535775589)

[4.5 Erste (1.) bis Dritte (3.) Normalform 14](#_Toc535775590)

[4.6 Redundanz 16](#_Toc535775591)

[4.7 Integrität 16](#_Toc535775592)

[4.8 Kardinalität 16](#_Toc535775593)

[4.9 Index 17](#_Toc535775594)

[4.10 Persönliches Fazit zu Kapitel 4 17](#_Toc535775595)

[5. DATENSCHUTZ 18](#_Toc535775596)

[5.1 Allgemeines 18](#_Toc535775597)

[5.2 Persönliches Fazit zu Kapitel 5 18](#_Toc535775598)

[6. DATENSICHERHEIT 19](#_Toc535775599)

[6.1 Allgemeines 19](#_Toc535775600)

[6.2 Persönliches Fazit zu Kapitel 6 19](#_Toc535775601)

[7. DATENKONSISTENZ 20](#_Toc535775602)

[7.1 Allgemeines 20](#_Toc535775603)

[7.2 Persönliches Fazit zu Kapitel 7 20](#_Toc535775604)

[8. DATENANALYSE 21](#_Toc535775605)

[8.1 Allgemeines 21](#_Toc535775606)

[8.2 Persönliches Fazit zu Kapitel 8 21](#_Toc535775607)

[9. MODELLIERUNG UND TRANSFORMATION 22](#_Toc535775608)

[9.1 Allgemeines 22](#_Toc535775609)

[9.2 Persönliches Fazit zu Kapitel 9 22](#_Toc535775610)

[Glossar 23](#_Toc535775611)

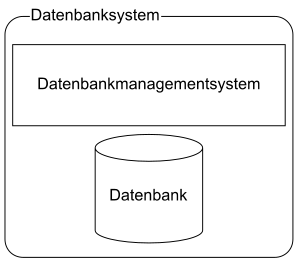
[Abbildungsverzeichnis 24](#_Toc535775612)

# GRUNDLAGEN

## 1.1 DB, DBS und DBMS

Kapitel 1.1 soll kurz die Grundbegriffe Datenbank (DB), Datenbanksystem (DBS) und Datenbankmanagementsystem (DBMS) erläutern – siehe Abbildung 1.1 unten.

Der Oberbegriff **Datenbanksystem** besteht dabei aus zwei Teilen, einem Datenbankmanagementsystem und einer Datenbank. Ein solches Datenbanksystem dient zur elektronischen Datenverwaltung bei grossen Datenmengen.

Ein **Datenbankmanagementsystem**(DBMS) ist eine Software, die für das Datenbanksystem installiert und eingerichtet wird.

Eine **Datenbank** ist ein elektronisches Verwaltungssystem, das besonders mit grossen Datenmengen effizient, widerspruchsfrei und dauerhaft umgehen muss sowie logische Zusammenhänge digital abbilden kann, wie z.B. ein Datenmodell.

Quelle: <http://www.datenbanken-verstehen.de/lexikon/datenbanksystem/>

Abbildung 1.1 – Schema Datenbanksystem

Ich möchte in diesem Dokument durchgehend ein fiktives Beispiel verwenden, um die einzelnen Begriffe und Kapitel näher zu erklären:

In diesem Beispiel handelt es sich um ein Unternehmen im Bereich der Unterhaltungselektronik mit dem fiktiven Namen «Blizzcon». Dieses Unternehmen hat 3 Computerspiele entwickelt und vertreibt diese auch.

«Blizzcon» möchte mit Hilfe eines solchen Datenbanksystems die Übersicht über die Spiele und deren Releases sowie vor allem deren Kundenbasis erhalten.

## **1.2 Aufgaben eines Datenbanksystems**

Von der Definition her unterstützt ein Datenbanksystem die computergestützte Datenverarbeitung von Informationen, die durch eine Datenapplikation erzeugt werden. Das Datenbanksystem strukturiert und speichert die Informationen dann in einer Datenbank.

Quelle : <http://www.datenbanken-verstehen.de/lexikon/datenbanksystem/>

In meinem Beispiel der Firma «Blizzcon» gibt es zwei zentrale Tabellen. Die erste Tabelle erfasst die Daten der von «Blizzcon» veröffentlichten Spiele (siehe Tabelle unten) mit der entsprechenden Versionsnummer und dem dazugehörigen Abo-Preis pro Monat in Euro.



Die zweite Tabelle beinhaltet Daten zu den Kunden der Firma «Blizzcon» (siehe Tabelle unten) unter Angabe des Wohnsitzes und Informationen zu allfälligen Abos.



Mit Hilfe dieser beiden Tabellen sollen nun Informationen über die Kundenbasis für alle Spiele verfügbar gemacht werden.

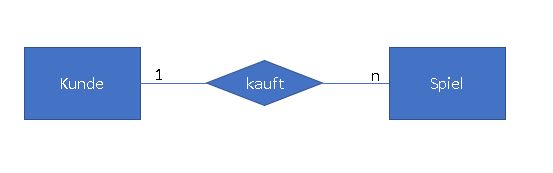
## 1.3 Entity-Relationship Modell (ERM)

Das Entity-Relationship Modell – abgekürzt ERM - dient als Grundlage für einen Datenbankentwurf. Bevor mit SQL gearbeitet wird (siehe Kapitel 2), wird zuerst mit einem ER-Modell geplant, wie die Datenbankstruktur aufgebaut werden soll.

Mithilfe des Entity-Relationship Modells findet eine Typisierung von Objekten, ihrerrelationalen Beziehungen untereinander und der zu überführenden Attribute statt.

Quelle : <http://www.datenbanken-verstehen.de/datenmodellierung/entity-relationship-modell/>

Hier ein Beispiel:



Das obige Beispiel zeigt eine 1:n Beziehung zwischen den beiden für «Blizzcon» wichtigen Entitäten Kunde und Spiel. Es handelt sich um eine 1:n Beziehung, da ein Kunde mehrere Spiele erwerben kann.

## 1.4 Entity-Relationship Diagramm (ERD)

Das Entity-Relationship Diagramm(ERD) ist ein Modell, das zur Modellierung von logischen Datenbankbeziehungen verwendet wird. Das ERD ist sehr weit verbreitet und wurde im Laufe der Zeit mehrmals ergänzt und überarbeitet. Der Vorteil des Modells ist seine einfache Übertragung in ein logisches, relationales Datenbankmodell - das sogenannte Relationenmodell von Codd.

Die Grundelemente einesjeden Entity-Relationship Diagramm bilden Entitäten (entities), Beziehungen und Attribute. Diese sind wie folgt definiert:

Eine **Entität** ist ein individuell identifizierbares Objekt der Wirklichkeit und wird im Diagramm als Rechteck dargestellt.

Eine **Beziehung** ist eine Verknüpfung / Zusammenhang zwischen zwei oder mehreren Entitäten. Sie wird als Raute dargestellt.

Ein **Attribut** ist eine Eigenschaft, die im Kontext zu einer Entität steht. Es wird als Ellipse dargestellt.

Quelle: <http://www.datenbanken-verstehen.de/lexikon/er-diagramm/>



Abbildung 1.2 – Grundelemente eines ERD

## 1.5 Relationale Datenbank

Das relationale Datenbankmodell ist das am weitesten verbreitete Datenmodell, welches in der Datenbankentwicklung als Standard genutzt wird. Das Fundament des Datenbankmodells besteht aus den folgenden vier Elementen:

* Tabellen
* Attribute
* Beziehungen
* Grundelemente der Algebra

Die relationale Datenbank stellt eine mathematische Beschreibung einer Tabelle und ihre Beziehung zu anderen möglichen Tabellen dar. Die Operationen auf diese Relationen werden durch die relationale Algebra bestimmt.

Quelle : <http://www.datenbanken-verstehen.de/datenbank-grundlagen/datenbankmodell/relationales-datenbankmodell/>

Ein Bild, das Text enthält.

Automatisch generierte Beschreibung

Abbildung 1.3 – Relationale Datenbank

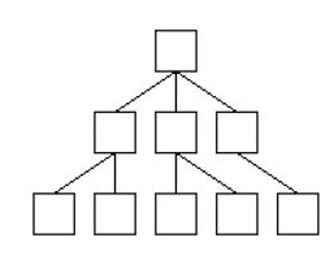
## 

## 1.6 Hierarchische Datenbank

Das hierarchische Datenbankmodell ist das älteste Datenbankmodell und wird als hierarchische Baumstruktur dargestellt**.** Viele Dateisysteme von Betriebssystemen benutzen dieses Modell, um Ihre Daten abzubilden. Heute ist dieses Datenbankmodell weitestgehend von anderen Modellen abgelöst worden.

Quelle : <http://www.datenbanken-verstehen.de/datenbank-grundlagen/datenbankmodell/hierarchisches-datenbankmodell/>

Wie man aus der Abbildung 1.4 unten sieht, sind die verschiedenen Böxchen über Eltern-Kind-Beziehungen miteinander verknüpft.



**Abbildung 1.4 – Hierarchische Datenbank**

**Der Entwurf einer hierarchischen Datenbank hat Vorteile und Nachteile:**

|  |  |
| --- | --- |
| **VORTEILE** | **NACHTEILE** |
| **Einfach zu verstehen** | **Unflexibel** |
| **Kurze Zugriffszeiten** | **Zugriffspfade müssen bekannt sein** |
| **Festgelegte Suchpfade** |  |

## 1.7 Persönliches Fazit zu Kapitel 1

Kapitel 1 vermittelt die Grundlagen zum Thema Datenbanksysteme und erläutert die Begriffe..

# DATENBANKSPRACHE

Als Datenbanksprache bezeichnet man eine Sprache, die für den Einsatz in Datenbanksystemen entwickelt wurde. Mit Hilfe einer solchen Datenbank-Sprache kann ein Programm oder ein Benutzer mit der Datenbank sozusagen «kommunizieren». Da ein wichtiger Teil der Arbeit mit Datenbanksystemen die Formulierung von Abfragen ist, gehört zum Sprachumfang in der Regel auch die Datenbank-[Abfragesprache](https://de.wikipedia.org/wiki/Abfragesprache).

Quelle : <http://www.datenbanken-verstehen.de/datenbank-grundlagen/dbms/datenbanksprachen/>

## 2.1 Structured Query Language (SQL)

**SQL** ist die am weitesten verbreitete Datenbank-Sprache und dient dazu, Definitionen, Abfragen und Manipulationen von Daten in relationalen Datenbanken durchzuführen. Man benützt die Sprache auch, um relationale Datenbanken zu verändern, Datensätze zu löschen oder einzufügen oder generell Datenbestände abzufragen.

Die Sprache SQL basiert auf der relationalen Algebra, ihre Syntax ist relativ einfach gebaut und semantisch an die englische Umgangssprache angelehnt.

Quelle : <http://www.datenbanken-verstehen.de/datenbank-grundlagen/dbms/dml-ddl-dcl-kommandos/>

## 2.2 Data Control Language (DCL)

Die **DCL** ist derjenige Teil einer Datenbanksprache, der verwendet wird um Berechtigungen / Rechte - zum Beispiel für ein Projekt - zu geben oder zu entziehen. Daher auch der Name Data-Control.

In meinem Beispiel wäre ein Einsatzbereich für DCL, wenn die Firma «Blizzcon» ihren Mitarbeitern Rechte zu den Datenbanken gibt, damit sie auch daran arbeiten können.

Quelle : <http://www.datenbanken-verstehen.de/datenbank-grundlagen/dbms/dml-ddl-dcl-kommandos/>

## 2.3 Data Definition Language (DDL)

Die **DDL** ist ein weiterer Teil der Datenbank-Sprache SQL und ermöglicht es, die Datenstrukturen und verwandte Elemente zu beschreiben, zu ändern oder zu entfernen. Ursprünglich bezog sich die Data Definition Language auf Datenbanksysteme.

Als Datenbanksprache ist die DDL die Datenbeschreibungssprache einer Datenbank.

Quelle : <https://de.wikipedia.org/wiki/Data_Definition_Language>

## Data Query Language (DQL)

Die **DQL** (auch Data Manipulation Language) ist ein Teil der SQL-Sprache, die sich mit allen Auswahl-/Einfüge-/Änderungs- und Löschoperationen beschäftigt. Besonders bei **Auswahloperationen** kann man enorme Unterschiede feststellen, die sich in ihrer Antwortgeschwindigkeit widerspiegeln.

So kann man z.B. mit dem Befehl SELECT in meinem Beispiel aus der Tabelle «KUNDE» der Firma «Blizzcon» nur die Spalten [SUBSCRIPTION] und [LAND] wie folgt abfragen:

SELECT [LAND], [SUBSCRIPTION] FROM [dbo] [KUNDE]

Das Resultat sieht entsprechend wie folgt aus:



Quelle : <http://www.datenbanken-verstehen.de/datenbank-grundlagen/dbms/dml-ddl-dcl-kommandos/>

## 2.5 Persönliches Fazit zu Kapitel 2

Kapitel 2 erklärt alle wichtigen Datenbanksprachen und Teile der Sprache SQL.

# 3. BEGRIFFE EINER DATENBANK

## 3.1 Entität

In einer Datenbank ist eine **Entitä**t ein konkretes Objekt bzw. ein konkreter Sachverhalt, der sich eindeutig von anderen Entitäten des gleichen Entitätstyps unterscheidet.

Quelle[[: http://www.datenbanken-verstehen.de/datenmodellierung/entitaeten/](http://www.datenbanken-verstehen.de/datenmodellierung/entitaeten/)](http://www.datenbanken-verstehen.de/datenmodellierung/entitaeten/)

In meinem Beispiel der Firma «Blizzcon» existieren zwei Entitäten – SPIEL und KUNDE.

## 3.2 Entitätsmenge

Von der Definition her repräsentiert eine **Entitätsmenge** alle Datensätze, die zu einer Entität gehören.

In der Tabelle «Kunde» ist KUNDE die Entität und damit ist die Entitätsmenge die Gesamtheit aller Daten zu dieser Entität, d.h. die gesamte Tabelle **«Kunde».**



## 3.3 Tabelle

Jede Datenbank hat mindestens 1 **Tabelle**, in der Regel aber mehrere. In Tabellen werden Daten gespeichert und jede Tabelle gruppiert bestimmte Daten.

Quelle : <http://sql.lernenhoch2.de/lernen/sql-einleitung/tabellen-erklart/>

In meinem Beispiel der Firma «Blizzcon» gibt es eine Tabelle „Kunde“, in der alle Kunden mit ihren spezifischen Daten (wie Name, Geburtsjahr, Land, etc.) gespeichert werden (siehe Abbildung unten).



## 3.4 Datensatz

Bei einem **Datensatz** (Tupel) handelt es sich um eine inhaltlich zusammenhängende Gruppe von Datenfeldern. Beispielsweise bezeichnen Name, Adresse, sowie Geburtsdatum und Geburtsort die Datenfelder zu einem Datensatz einer Person.

Ebenso handelt es sich bei Artikelnummer, Artikelname, ID und Preis um Datenfelder eines Datensatzes eines Artikels. Hieraus ist ersichtlich, dass Datensätze wichtige Informationen zu einer bestimmten Kategorie zusammenfassen.

In der heutigen Zeit werden Datensätze vornehmlich in elektronischer Form, beispielsweise in Datenbanken, verwendet.

Quelle : <https://www.it-service24.ch/lexikon/d/datensatz/>



Im obigen Beispiel wäre z.B. die markierte Zeile ein Datensatz in der Tabelle SPIEL.

## 3.5 Attribut

Ein **Attribut** ist ein Merkmal eines Tupels und beschreibt eine Eigenschaft einer Entitätsmenge. In einem relationalen Datenbankmodell ist ein Attribut eine Spalte einer Tabelle.

Eine Eigenschaft eines Attributs ist ein konkreter Attributwert, der ein Attribut beschreibt und durch einen numerischen, booleschen oder „Zeichenfolgenliteral“ repräsentiert wird.

Quelle : <http://www.datenbanken-verstehen.de/lexikon/attribut/>



Im obigen Beispiel entspricht die Spalte «Release» einem Attribut.

## 3.6 Feld

Ein **Feld** ist eine benannte Einheit an Information. Jeder Eintrag in einer Datenbank kann mehrere Felder unterschiedlicher Typen beinhalten, z.B. ein Textfeld mit der Bezeichnung "Preis" (um den Preis eines Spiels einzutragen) oder ein Auswahlmenü "Spiele" welches erlaubt, aus einer vorgegebenen Liste aller Spiele eines auszuwählen (vorausgesetzt, diese Liste wurde nur einmal angelegt). Durch die Kombination mehrerer Felder mit geeigneten Namen und Typen sollte es möglich sein, alle notwendigen Informationen über ein beliebiges Thema in der Datenbank zu erfassen.

Quelle : <https://docs.moodle.org/19/de/Datenbank-Felder>



Im obigenBeispiel wäre z.B „Name“ ein Feldwert in der Tabelle SPIEL.

## 3.7 Wert /Wertbereich

Der **Wert** ist sozusagen immer der Wert in einer Tabelle, z.B in der untenstehenden Abbildung die «Bewertung».



## 3.8 Der Begriff Null

Den **Nullwert** bezeichnet man in der Informatik als einen Zustand, der das Fehlen eines Wertes anzeigen soll.



In der obigen Abbilung fehlen zum Beispiel bei zwei Datensätzen Angaben zum Geburtsjahr bzw. Land.

## 3.9 Persönliches Fazit zu Kapitel 3

Kapitel 3 erklärt wichtige Begriffe, wie Entität, Tabelle, Datensatz, Attribut, etc., welche man verstehen muss, um Texte über Datenmodellierung zu verstehen.

# 4. RELATION & SCHLÜSSEL

## 4.1 Beziehung (Relation)

Eine Relation legt lediglich fest, ob ein Tupel zu ihr gehört oder nicht. Eine Relation hat die Form einer Tabelle, und ist demnach eine [Verknüpfung](https://www.itwissen.info/join-Verknuepfung.html) von [Zeilen](https://www.itwissen.info/Zeile-line.html) und Spalten.

Im Hinblick auf das [Entity-Relationship Modell](https://www.itwissen.info/ERM-entity-relationship-model-ER-Modell.html) werden sowohl dessen [Entities](https://www.itwissen.info/Instanz-entity.html) als auch Beziehungen selbst als Relationen modelliert. Aus Sicht einer Datenbank bezieht sich eine Relation nicht auf die Beziehung zwischen Tabellen. Vielmehr besteht ein relationales [Datenbankmodell](https://www.itwissen.info/Datenbankmodell-database-model.html) aus einer Anzahl von Relationen, in denen dann logisch zusammengehörende [Daten](https://www.itwissen.info/Daten-data.html) gespeichert werden.

Quelle : <https://www.itwissen.info/Relation-relation.html>

## 4.2 Primärschlüssel

Der **Primärschlüssel** kommt in relationalen Datenbanken zum Einsatz und wird zur eindeutigen Identifizierung eines Datensatzes verwendet. In einer normalisierten Datenbank besitzen alle Tabellen einen Primärschlüssel.

Der Wert eines Primärschlüssels muss in einer Tabelle einmalig sein, da er jeden Datensatz eindeutig kennzeichnet. Des Weiteren wird er häufig als [Datenbank-Index](http://www.datenbanken-verstehen.de/datenmodellierung/datenbank-index/) verwendet, um die Daten auf der Festplatte abzulegen.

Quelle : <http://www.datenbanken-verstehen.de/datenmodellierung/primaerschluessel/>

## 4.3 Fremdschlüssel

Der **Fremdschlüssel** kann Bestandteil einer Tabelle in einer relationalen Datenbank sein. Dabei handelt es sich um eine Schlüsselspalte, die auf einen [Primärschlüssel](http://www.datenbanken-verstehen.de/datenmodellierung/primaerschluessel/) einer anderen oder aber derselben Tabelle verweist.

Es kann sich dabei um einen einfachen oder zusammengesetzten Schlüssel handeln. Das hängt davon ab, wie der Primärschlüssel der Tabelle aufgebaut ist.

Aufgrund der [referentiellen Integrität](http://www.datenbanken-verstehen.de/datenmodellierung/referentielle-integritaet/) kann der Fremdschlüssel nur Werte annehmen, die in der Tabelle vorhanden sind. Zudem kann eine beliebige Anzahl von Datensätzen den gleichen Fremdschlüsselwert aufweisen.

Quelle : <http://www.datenbanken-verstehen.de/datenmodellierung/fremdschluessel/>

## **4.4 Normalisierung**

Unter **Normalisierung** eines relationalen Datenbankmodells versteht man die Aufteilung von Attributen in mehrere Relationen (Tabellen) mithilfe der Normalisierungsregeln und deren Normalformen, sodass eine Form entsteht, die keine vermeidbaren Redundanzen mehr enthält.

Ziel der Normalisierung ist, eine redundanzfreie Datenspeicherung zu erstellen. Damit eine Tabelle redundanzfrei ist, müssen alle wiederholten und unnötigen Daten entfernt werden, ohne dass es zu Informationsverlusten kommt.

Quelle : <http://www.datenbanken-verstehen.de/datenmodellierung/normalisierung/>

## Erste (1.) bis Dritte (3.) Normalform

1.Normalform

Informationen, die vorher unstrukturiert und unsortiert vorlagen, werden nun einheitlich und klar strukturiert.

Es bedeutet, dass jede Information innerhalb einer Tabelle eine eigene Tabellenspalte bekommtund zusammenhängende Informationen, wie zum Beispiel die Postleitzahl (PLZ) und der Ort, nicht in einer einzigen Tabellenspalte vorliegen dürfen.

Hier ein Beispiel:



Nach der Verwendung der 1. Normalform sieht die Tabelle so aus:



Damit die Datenbank die Tabelle besser versteht, muss man sie in mehrere Teile zerlegen.

2.Normalform

Die zweite Normalform ist ein wichtiger Schritt zu einer voll normalisierten relationalen Datenbank. Sie prüft, ob eine vollständige funktional oder nur eine funktionale Abhängigkeit von Werten zu einer bestimmten Teilmenge existiert.

Die zweite Normalform wird meistens schon indirekt erreicht, wenn der Datenbankentwickler mit der Erstellung eines [ER-Diagramms](http://www.datenbanken-verstehen.de/lexikon/er-diagramm/) beschäftigt ist. Die logische Aufspaltung von komplexen Sachverhalten zwingt den Datenbankentwickler Geschäftsprozesse in Relationen abzubilden.



Nach der Verwendung der 2. Normalform sieht die Tabelle wie folgt aus:





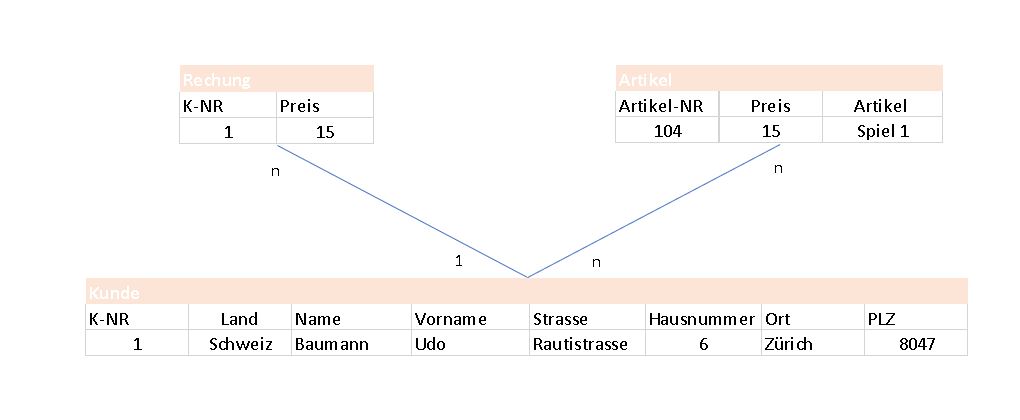
Die Spalten, die von einem Schlüsselkandidaten nicht vollständig funktionalabhängig sind, werden in einer Untertabelle ausgelagert. Der Teil des Schlüsselkandidaten, von dem eine ausgelagerten Spalte funktional abhängig ist, wird Primärschlüssel der neuen Tabelle. In der zweiten Normalform werden auch die ersten [Beziehungen in Datenbanken](http://www.datenbanken-verstehen.de/datenmodellierung/beziehungen-datenbanken/) festgelegt.

3.Normalform

Die dritte Normalform ist das Ziel einer erfolgreichen **Normalisierung** in einem relationalen Datenbankmodell.

Sie verhindert einerseits Anomalien und Redundanzen in Datensätzen und andererseits bietet sie genügend Performance für SQL-Abfragen.

Die dritte Normalform ist oft ausreichend, um die perfekte Balance aus Redundanz, Performance und Flexibilität für eine Datenbank zu gewährleisten.

Quelle : <http://www.datenbanken-verstehen.de/?s=1NF>

Die 3. Normalform ist eigentlich fast genau gleich wie die 2., ausser dass man bei der dritten Normalform die Tabellen detaillierter aufzeichnet und sie nach Entitäts-Beziehung mit einander verbindet.

## 4.6 Redundanz

**Redundanzen** sind **unnötige Informationen**, die man ohne jeglichen Informationsverlust weglassen kann.



In diesem Beispiel sieht man, dass Klaus Müller zwei Mal in der Tabelle vorkommt. Fast alle Informationen sind gleich, ausser das «Land». Nach einer Prüfung, ob es sich wirklich um den «gleichen» Klaus Müller» handelt, kann man die veraltete Zeile weglassen, da sie nur alte Informationen enthält.

## 4.7 Integrität

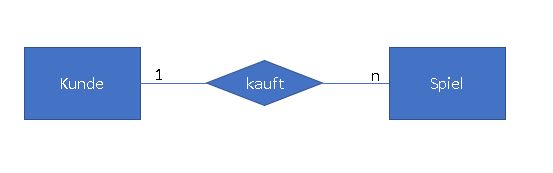
Die **Datenintegrität/Integrität** kann durch sogenannte Constraints geregelt werden. Diese Regeln bestimmen im DBMS, wie die Daten verändert werden dürfen.

Der wichtigste Vertreter bei relationalen Datenbanken ist der [Fremdschlüssel](http://www.datenbanken-verstehen.de/datenmodellierung/fremdschluessel/), auch Foreign Key Constraint genannt. Dieser Constraint verhindert zum Beispiel, dass Daten gelöscht werden können, die noch in anderen Tabellen referenziert sind.

Quelle : <http://www.datenbanken-verstehen.de/datenbank-grundlagen/dbms/integritaet/>

## 4.8 Kardinalität

Die Kardinalität zwischen dem Entitätstyp 1 und dem Entitätystyp 2 gibt an, wie viele Entitäten des Entitätstyps 2 höchstens mit einer Entität des Entitätstyps 1 in Beziehung stehen.



1. höchstens eine Entität, (n) mehrere Entitäten

Quelle : <https://www.tinohempel.de/info/info/datenbank/kardinalitaet.htm>

## 4.9 Index

Der **Datenbankindex** ist eine **Datenstruktur** mit deren Hilfe die [Abfrageoptimierung](http://www.datenbanken-verstehen.de/datenbank-grundlagen/dbms/datenbank-optimierung/) gesteigert werden kann. Mittels einer Indextabelle werden die Daten sortiert auf dem Datenträger abgelegt.Der Index selbst stellt einen Zeiger dar, der entweder auf einen weiteren Index oder auf einen Datensatz zeigt. Dadurch findet eine Trennung von Daten- und Index-Strukturen statt.

Quelle : <http://www.datenbanken-verstehen.de/datenmodellierung/datenbank-index/>

## 4.10 Persönliches Fazit zu Kapitel 4

Das Kapitel 4 erklärt wichtige Begriffe wie Schlüssel und die Normalisierung.

# 5. DATENSCHUTZ

## 5.1 Allgemeines

Der Datenschutz ist dafür da, bestimmte Informationen von Menschen oder Dingen zu schützen. So darf ein Arbeitgeber beispielsweise keine privaten Mails von Mitarbeitern lesen oder die privaten Telefonate aufzeichnen.

Das erste Ziel des Datenschutzes ist es, das informationelle Selbstbestimmungsrecht des Menschen zu verteidigen. Diese Aufgabe ist nicht immer einfach, da es zum Teil auch erlaubte Interessen geben kann, die dieses Selbstbestimmungsrecht einschränken, so etwa bei polizeilichen Ermittlungen.

Der Datenschutz soll garantieren, dass immer nur so viele persönliche Daten wie nötig gesammelt und bearbeitet werden, und dass man als betroffene Person auch die Möglichkeit hat, die Bearbeitung der Daten über sich so weit wie möglich zu kontrollieren und allenfalls zu verhindern. Zu diesem Zweck schreibt das Datenschutzgesetz ein Auskunftsrecht fest, das bei den Inhabern von Datensammlungen benützt werden darf.

Quelle : <http://www.lernender.ch/files/Leistungsziele/Branchen03/Bank/Files/2_4_2_2_Datenschutz_erklaeren.pdf>

Seit fast einem Jahr gibt es neu auch die europäische Datenschutz-Verordnung. Sie ist eine [Verordnung der Europäischen Union](https://de.wikipedia.org/wiki/Verordnung_(EU)), mit der die Regeln zur Verarbeitung [personenbezogener Daten](https://de.wikipedia.org/wiki/Personenbezogene_Daten) durch private Unternehmen und öffentliche Stellen [EU](https://de.wikipedia.org/wiki/Europ%C3%A4ische_Union)-weit vereinheitlicht werden. Dadurch soll einerseits der Schutz personenbezogener Daten innerhalb der [Europäischen Union](https://de.wikipedia.org/wiki/Europ%C3%A4ische_Union) sichergestellt, und auch andererseits der freie [Datenverkehr](https://de.wikipedia.org/wiki/Datenverkehr) innerhalb des [Europäischen Binnenmarktes](https://de.wikipedia.org/wiki/Europ%C3%A4ischer_Binnenmarkt) gewährleistet werden. Bei Nicht-Einhaltung dieser Verordnung drohen hohe Geldstrafen.

Neu kann jeder Kunde, z.B auf der Website eines Unternehmens, entscheiden ob die eigenen persönlich Daten weitergegeben werden können oder nicht.

In meinem Beispiel der Firma «Blizzcon» kann Udo Baumann, wenn er sich auf der Homepage der Firma registriert, auswählen, ob seine Daten privat oder öffentlich gestellt werden sollen.

## 5.2 Persönliches Fazit zu Kapitel 5

Kapitel 5 erklärt, was Datenschutz bedeutet und warum es wichtig ist.

# 6. DATENSICHERHEIT

## 6.1 Allgemeines

Als Datensicherheit oder Informationssicherheit bezeichnet man die Eigenschaften von IT-Systemen, welche die Vertraulichkeit, Verfügbarkeit und die Integrität der Informationen sicherstellen. Die Informationen werden dabei eingeschränkt (Vertraulichkeit) überprüfbar gemacht (Integrität) und mehrfach abgespeichert (Verfügbarkeit).

Zu den Möglichkeiten, die zur Datensicherheit beitragen, zählen also:

* Verschlüsselung der Daten (Geheimschrift)
* Digitale Signaturen (elektronisches Unterschreiben)
* Zugriffsberechtigungen
* DRM (Digital Rights Management - also etwa digitale Rechteverwaltung)
* Backup
* Antiviren Software
* Firewalls
* Protokollierung

Die Datensicherheit kann den Datenschutz zwar unterstützen, indem personenbezogene Daten vor unberechtigtem Zugriff geschützt werden. Aber gerade Backups und die Speicherung von Daten in (unverschlüsselten) Cloud-Speichern bergen für den Datenschutz ein Risiko und haben damit auch ein Missbrauchspotential.

Quelle : <https://www.mastertools.ch/blog/posts/2013/september/datensicherheit-vs-datenschutz/>

## 6.2 Persönliches Fazit zu Kapitel 6

Kapitel 6 beschreibt die Bedeutung der Datensicherheit und zählt Möglichkeiten auf, diese zu gewährleisten.

# 7. DATENKONSISTENZ

## 7.1 Allgemeines

Datenkonsistenz, auch Datenintegrität genannt, bezeichnet die Korrektheit der in einer Datenbank oder einem verteilten System gespeicherten Daten.

Konsistente Daten sind widerspruchsfreie Daten.

Aus technischer Sicht führen inkonsistente Informationen in einer Datenbank zu teilweise schwerwiegenden Fehlern, wenn die darüber liegende ausführende Anwendung auf Inkonsistenzen nicht vorbereitet ist.

Aus der Anwendersicht bedeuteten Daten-Inkonsistenzen eine mangelnde Datenqualität. Diese führt zu erhöhten Kosten, Mehraufwand und eventuell unzufriedenen Kunden.

Quelle : <http://www.datenbanken-verstehen.de/lexikon/datenkonsistenz/>

## 7.2 Persönliches Fazit zu Kapitel 7

Kapitel 7 erklärt den Begriff Datenkonsistenz und warum sie aus technischer Sicht und aus Sicht des Benutzers von Bedeutung ist.

# 8. DATENANALYSE

## 8.1 Allgemeines

Die explorative Datenanalyse wird auch als explorative Statistikbezeichnet und bildet ein Teilgebiet der Statistik. Es werden Daten analysiert, zu denen oft kaum oder sogar keine bekannten Zusammenhänge bestehen.

Dabei besteht das Ziel der explorativen Datenanalyse darin, einen Überblick über die vorliegenden Daten zu erhalten, um diese besser einschätzen und bewerten zu können. Erst im Rahmen der Analyse fallen Regelmäßigkeiten, Abhängigkeiten oder besondere Zusammenhänge zwischen scheinbar vollkommen ungleichen Daten auf.

Wurden ausreichend Muster und Zusammenhänge entdeckt, um eine Aussage zu treffen, wird anschließend gezielt nach Ausreißern in dieser Masse gesucht. Die einzelnen Ausreißer werden wiederum analysiert, um zu entscheiden, ob sie aus dem Datenpool entfernt werden können oder sogar die Analyse noch weiter verfeinert werden muss.

**Vorteile:**

* Durch die wiederholte Analyse werden die Zusammenhänge der Daten immer wirklichkeitsgetreuer dargestellt.
* Die explorative Datenanalyse eignet sich besonders gut als Lern- oder Erkenntniswerkzeug.
* Es können viele Informationen zugleich transportiert und transparent gemacht werden.

**Nachteile:**

* Der Analytiker benötigt einen umfassenden Erfahrungsschatz, um realistische Zusammenhänge abzubilden.
* Die selektive Wahrnehmung des Analytikers spielt in jeder explorativen Datenanalyse eine Rolle.
* Die Fragestellungen sind für manche Personen unter Umständen nicht sinnvoll und können daher nur schwer aussagekräftig bewertet werden.

## 8.2 Persönliches Fazit zu Kapitel 8

Kapitel 8 erklärt die Datenanalyse und wo man sie benötigt.

# 9. MODELLIERUNG UND TRANSFORMATION

## 9.1 Allgemeines

Um erfolgreich Daten zu modellieren, ist es besonders wichtig, die Grundlagen und Zusammenhänge zwischen den einzelnen Themen zu verstehen und anhand von Beispielen nachzubilden.

Viele Datenbankentwickler sprechen von einer guten Datenmodellierung, wenn folgende Punkte erreicht sind:

* redundanzfreie Datenspeicherung
* hohe Datenkonsistenz.

Eine **redundanzfreie Datenspeicherung** liegt dann vor, wenn jede Information in einer Datenbank genau einmal vorkommt.

Des Weiteren muss eine **hohe Datenkonsistenz** verfolgt werden, so dass Daten eindeutige Informationen darstellen.

Mit Datenmodellierung bezeichnet man in der [Informatik](https://de.wikipedia.org/wiki/Informatik) Verfahren zur formalen Abbildung der in einem definierten Kontext relevanten [Objekte](https://de.wikipedia.org/wiki/Entit%C3%A4t_(Informatik)) mit ihren [Attribute](https://de.wikipedia.org/wiki/Attribut_(Objekt))n und Beziehungen.

Das eigentliche Ziel einer Datenmodellierung ist die Definition und Spezifikation der in einem [Informationssystem](https://de.wikipedia.org/wiki/Informationssystem) zu verwaltenden Objekte, ihrer für die Informationszwecke erforderlichen Attribute und der Zusammenhänge zwischen den [Informationsobjekten](https://de.wikipedia.org/wiki/Entit%C3%A4t_(Informatik)), um so einen guten Überblick über die Datensicht des Systems erhalten zu können.

Bei einer Datenmodell-Transformation geht es darum, einen Datenbank Entwurf z.B ERM zu einer Datenbank zu transformieren. Man besitzt Daten und setzt ein Modell in eine Datenbank um. Dafür gibt es verschiedene Programme auf dem Computer wie zum Beispiel Microsoft Access.

## 9.2 Persönliches Fazit zu Kapitel 9

Kapitel 9 erklärt was eine Datenmodellierung und -Transformation ist.

# Glossar

|  |  |
| --- | --- |
| **Begriff** | **Erklärung** |
|  |  |
| Syntax | Regeln der Sprachstruktur |
| Tupel | Synonym von Datensatz |
| boolesch | Variable in der Algebra |
|  |  |
|  |  |

# Abbildungsverzeichnis

Abbildung 1.1 – Schema Datenbanksystem

Abbildung 1.2 – Grundelemente eines ERD

Abbildung 1.3 – Relationale Datenbank

Abbildung 1.4 – Hierarchische Datenbank