LS 3-1: Datenbankmodell auswählen

# Situation

Die Banor-IT als Ihr Ausbildungsbetrieb bekommt einen Auftrag von Höllental-Senfmühle einen Online-Shop zu implementieren.



Die Höllental-Senfmühle im Sauerland /in Bayern durch die Besinnung auf gutes Essen im Rahmen der Corona-Pandemie immer mehr Anfragen sowohl von Privatpersonen als auch von Restaurants und Händlern. Im Angebot ist Senf unterschiedlicher Konsistenz und unterschiedlicher Geschmacksrichtungen. Die kleine Senfmühle expandiert.

Nach Angaben der Inhaber Carola König und Maximilian Kaiser verwaltet das Unternehmen die Stamm- und Bewegungsdaten in Excel-Listen. Herr Kaiser stellt erfreut fest: „Das Geschäft geht sowas von durch die Decke, dass wir mit unserem Datenmanagement hinterherhinken! Die Zunahme des Kundenkreises und der rasant wachsenden Zahl von Bestellungen veranlasst uns eine neue Plattform zu schaffen. Stellen Sie sich vor: Manche Aufträge wurden gar nicht oder falsch ausgeliefert.“

Das Shop-System soll selbst entworfen und programmiert werden. Für dieses Projekt bildet die Banor-IT mehrere Projektgruppen. Ein wesentlicher Teil wurde bereits in den vorhergehenden Situationen vorbereitend entwickelt. Sie dürfen das Projektteam „Datenbank“ unterstützen und nehmen dazu an der Kick-Off-Veranstaltung teil.

Welches Problem/Aufgabe stellt sich dar?

Welche Lösungsmöglichkeit(n) besteht(en)?

# Handlungsaufträge

1.Informieren Sie sich über die Grundlagen der Datenbanken und beantworten Sie die Impulsfragen.

Info: Lastenheft Info: Grundlagen DB

2.Präsentieren Sie Ihre Ergebnisse dem Plenum.

3.Diskussion: Welches Datenbankmanagementsystem würden Sie für das Projekt empfehlen?

Begründen Sie Ihre Entscheidung im Plenum!

Impulsfragen zu den Grundlagen von Datenbanken

**AA**

1. Für den Online-Shop muss entschieden werden, welches Datenbankmodell zur Anwendung kommt. Zur Auswahl steht eine relationale Datenbank bzw. eine NoSQL-Datenbank.
2. Welche Vor- und Nachteile unterscheiden die beiden Datenbankmodelle?
3. Für welches Datenbankmodell würden Sie sich entscheiden? Begründen Sie Ihre Entscheidung unter Einbezug der Informationen aus dem Lastenheft von Höllental-Senfmühle!
4. Ein Datenbanksystem verfolgt verschiedene Ziele. Zum Teil beeinflussen sich diese Ziele Gegenseitig, weshalb nicht immer alle Ziele erfüllt werden können.

Welche Ziele, sind insbesondere für einen Online-Shop wesentlich? Begründen Sie!

1. Für den Online-Shop muss außerdem noch entschieden werden, mit welchem Datenbankmanagementsystem die Daten verwaltet werden sollen.
2. Wie hängen die Begriffe Datenbank, Datenbanksystem und Datenbankmanagementsystem zusammen?
3. Welche marktgängigen DBMS gibt es aktuell?
4. Um bei einem Online-Shop auf die Datenbank zugreifen zu können, bedarf es eines Client-Server-Modell.

Was ist unter einem Client-Server-Modell in Zusammenhang mit einem Datenbanksystem zu verstehen?

Lastenheft des Online-Webshops

**Anforderungen des Auftraggebers:**

**Zu erbringende Leistungen**

Erstellung eines Webshops mit einem ansprechenden, modernen Design

Topaktuell, der Zielgruppe folgend à Zielgruppendefinition ist durchzuführen, Corporate Identity, absolute Rechtssicherheit des Shops mit Berücksichtigung von Datenschutzgesetzen und Darstellung der Datenschutzbestimmungen, Tests der Produktivumgebung, Mobile Commerce first, kreativer Warenkorb

Technik

* Entwicklung einer eigenen Datenbank, bevorzugt Open Source, Programmierung, Implementierung des Webshops abhängig vom Hosting (Server im Unternehmen, managed Hosting, Cloud, etc.), SSL Zertifikat, Benutzer freundlich, barrierefrei, funktional, intuitiv, mehrsprachig DE -ENG, responsives Webdesign, leichte Wartung der Software

Datenmanagement

* Import der Stammdaten (z.B. Kunden-, Produkt- und Belegdaten), Kategorienmanagement, leichte Pflege und Wartung, vielfältige Auswahl an Zahlungssystemen und Versandoptionen, Festlegung von Standards für Zahlung und Lieferung mit Nachverfolgung, Steuersätze

Shopstatistik, Rechnungscenter, Lagerverwaltung, Suchfunktion

* Kundengruppenmanagement, Rabatte, Kundenrezensionen, Newsletter- und Gutscheinsystem

Produkte

* Empfehlungen für Produktauswahl (neueste und meist verkaufte Produkte, Aktionsprodukte) und Umsetzung erwünscht, Bilder in Absprache mit Kunden, Produktseiten mit kreativen Produktbeschreibungen und Storytelling, voll SEO-optimiert, Cross Selling

Retourenmanagement

Stornofunktion, um Retouren zu senken, nachhaltiges Retourenmanagement

Grundlagen Datenbanken

# Grundlagen Datenbanken Definition des Datenbankbegriffs

Ein Versuch, den Begriff Datenbank zu „definieren“ hat Ralf Adams[[1]](#footnote-2) vorgeschlagen: Unter einer Datenbank im engeren Sinn versteht man eine Informationssammlung, die so strukturiert ist, dass sie **zweckgebunden** und **effizient IT-gestützt** **verwaltet** und **ausgewertet** werden kann. (Adams, 2020, S. 5)

Typische Beispiele für die Anwendung von Datenbanken sind Personalverwaltung, Lagerwirtschaft oder Bestell- und Rechnungswesen. Die notwendigen Daten werden hier mit einem Datenbanksystem verwaltet. Weitere typische Einsatzgebiete sind Webanwendungen wie E-Shops oder zentrale Nachschlage- und Informationsverzeichnisse, wie z. B. Wikipedia.

# Struktur einer Datenbank

Daten sind vielfältig und werden in unterschiedlichen Situationen und zu unterschiedlichen Zwecken gesammelt und ausgewertet. Zur Datenorganisation, wird i. d. R. eine Datenbank herangezogen. So vielfältig die Daten sind, so vielfältig sind auch die Möglichkeiten, die Daten zu strukturieren. Je nach Bestimmungszweck und Anforderungen werden unterschiedliche Datenbankmodelle[[2]](#footnote-3) genutzt: Netzwerkdatenbanken, hierarchische, spaltenorientierte, dokumentenorientierte oder objektorientierte Datenbanken. Neben diesen eher spezifischen Datenbankmodellen gibt es zwei Modellarten, das einerseits weit verbreitet ist (relationale Datenbanken) und andererseits mit riesigen Datenmengen umgehen kann (NoSQL):

**Relationale Datenbanken:** Relationale Datenbanken sind am weitesten verbreitet. Die Daten werden in Tabellenform gespeichert, in sogenannten Relationen (mathematische Ausdrucksweise). Zwischen den Relationen (Tabellen) können Beziehungen definiert werden.

**NoSQL:** Mit den immer größer werdenden Datenmengen, haben relationale Datenbanken Probleme. Diese treten z. B. dann auf, wenn es zu einer extrem großen Anzahl gleichzeitiger Schreib- und Lesezugriffe oder Datenänderungen kommt. Des Weiteren macht die starke Abhängigkeit an ein bestimmtes Schema die relationalen Datenbanken unflexibel. Schemaänderungen großer Datenbanken sind sehr zeitaufwendig und im Hinblick auf einen möglichen Datenverlust nicht ungefährlich. NoSQL-Datenbanken besitzen diese starke Abhängigkeit von einem Schema nicht. Typisch für NoSQL-Datenbanken ist die Unterstützung von verteilten Datenbanken auf mehreren Servern. Vorreiter der Bewegung der NoSQL-Datenbanken waren beispielsweise Google, Amazon, eBay und Facebook. NoSQL-Datenbanken bieten alternative Datenstrukturen, greifen aber dennoch bewährte Muster auf, daher der Name NoSQL = Not only SQL.

# Aufbau und Organisation eines Datenbanksystems (DBS)

Eines der wichtigsten Ziele eines DBS ist die Datenunabhängigkeit. Diese wird durch die Trennung der physischen Speicherung der Daten und deren Verwaltung von den Anwendungsprogrammen erreicht. Zum einen sollte eine physische Datenunabhängigkeit bestehen. So kann die Struktur der gespeicherten Daten geändert werden, ohne dass die Anwendungsprogramme geändert werden müssen. Zum anderen ist auch die logische Datenunabhängigkeit eine wichtige Anforderung an ein Datenbanksystem. D. h. das Schema, wie die Daten in der Datenbank gespeichert sind (logische Datenbankstruktur) ist unabhängig von der anwenderspezifischen Sicht.

## 3-Ebenen-Modell (ANSI)

Um das zu erreichen, werden Datenbanksysteme nach dem 3-Ebenen-Modell nach ANSI-SPARC 1978 (American National Standards Institute/Standards Planning and Requirements Committee) konzipiert. Darunter versteht man die Möglichkeit des Datenzugriffs durch Programme oder Anwender, ohne dass diese über Kenntnisse der tatsächlichen technischen Umsetzung der physischen Speicherung der Daten verfügen.

**Externe Ebene**

beschreibt, wie

der Benutzer die

Daten sieht

(= Benutzersicht)

**Konzeptionelle Ebene**

beschreibt, wie

die Daten

strukturiert sind

(= logische Sicht)

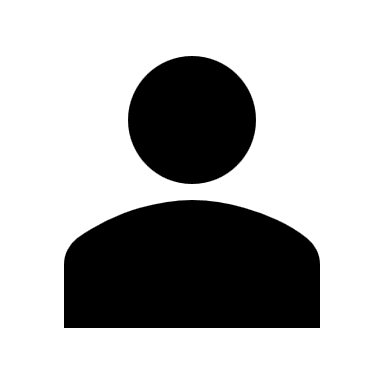
**Interne Ebene**

beschreibt, wo

die Daten

gespeichert sind

(= physische Sicht)



User 1

Anwendung 1

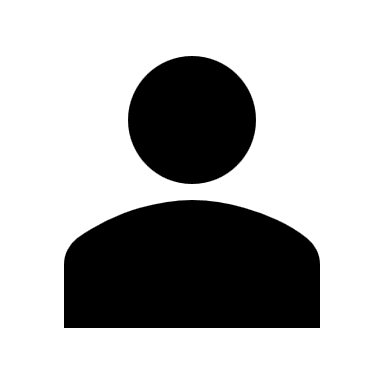
Beschreibung des Datenmodells

(logische Datenstruktur

Physische Datenstrukturen

(physikalischer Speicherort)

Anwendung n



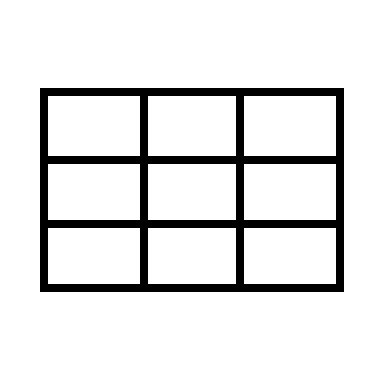
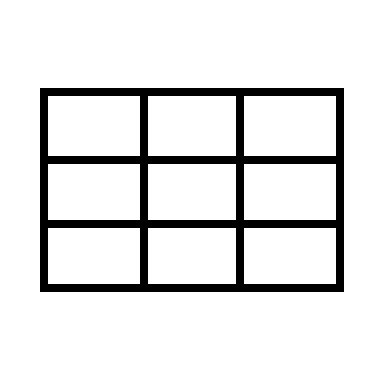
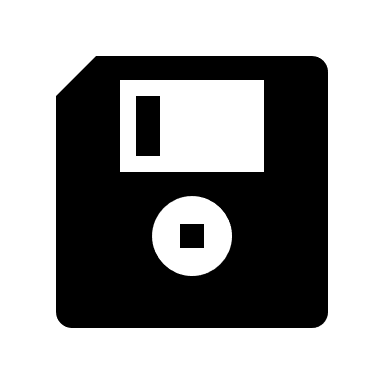
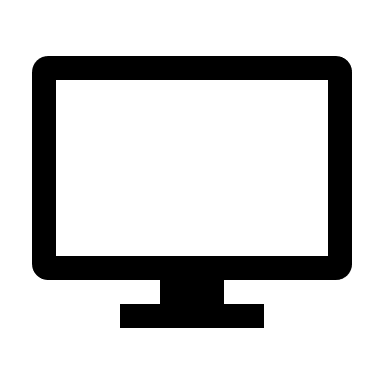
User n

z. B. Abfragen über

ein DMBS

Tabellen

C:\Daten\



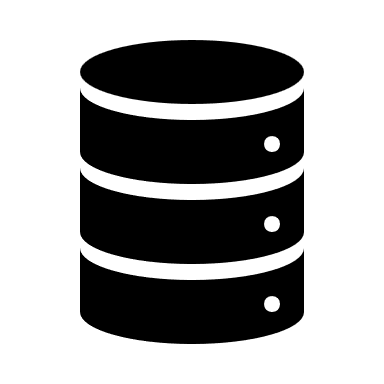
## Datenbanksystem

Das Datenbanksystem umfasst zwei Hauptkomponenten:

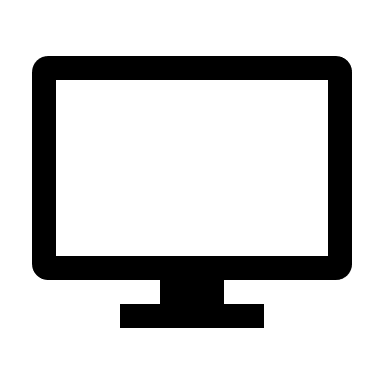
1. Datenbasis: stellt die Daten in einer Datenbank zur Verfügung (z. B. Tabellen)
2. Datenbankmanagementsystem: Software zum Verwalten der Daten in der Datenbasis  
   (z. B. dBase, Oracle, Paradox, MS-Access)

Ein Datenbanksystem lässt sich demnach folgendermaßen visualisieren:

**Datenbanksystem**



Datenbank n



Datenbank-  
managementsystem (DBMS)

Ein Datenbanksystem ermöglicht dem User – über ein DBMS – folgendes:

* **Datendefinition** (SQL-DDL): die Struktur einer Datenbasis aufzubauen (Datendefinition),

**Datenmanipulation** (SQL-DML): Daten pflegen, d. h. Datensätze eingeben, ändern und löschen

* **Datenabfrage** (SQL-DQL): Informationen aus der Datenbasis gewinnen

**Datenkontrolle** (SQL-DCL): Zugangs- und Zugriffsrechte verwalten

* **Datenübertragung**: Daten sichern, exportieren und importieren

# Ziele eines Datenbanksystems

Unternehmen organisieren Daten, um im einfachen Fall gezielte Informationen für den Arbeitsalltag herauszufiltern (z. B. Personaldaten abfragen) oder im komplexen Fall Zusammenhänge zu erschließen (z. B. Kundendaten auszuwerten und individuelle Angebote zu erstellen).

Hierfür ist es notwendig, dass Datenbanken bestimmte Kriterien erfüllen. Davon abgeleitet, ergeben sich folgende Ziele bei der Datenorganisation:

1. **Datenintegrität** sicherstellen

Die Daten müssen vollständig, korrekt und widerspruchsfrei sein.

1. **Redundanzfreiheit** gewährleisten

Jedes Datenelement sollte möglichst nur einmal gespeichert werden, z. B. sollte die Kundenanschrift nicht mehrfach abgelegt sein, weil bei einer Änderung alle Datensätze geändert werden müssen. Hierbei kann die Datenintegrität verletzt werden.

1. **Datenunabhängigkeit** sicherstellen

Unabhängigkeit vom Anwendungsprogramm: Die Daten sind unabhängig vom erzeugenden oder benutzenden Anwendungsprogramm.

Unabhängigkeit der logischen von der physischen Datenorganisation: Der Benutzer muss nur die Datenstrukturen kennen. Methoden zum Suchen, Ändern, Einfügen und Löschen von Datensätzen werden vom Datenbankmanagementsystem zur Verfügung gestellt.

Physische Datenunabhängigkeit: Das Datenbankmanagementsystem steuert und überwacht die peripheren Geräte, blockt bzw. entblockt Sätze, kontrolliert Überlaufbereiche, belegt Speicherräume oder gibt sie frei usw.

1. **Mehrfachzugriff** erlauben

Jeder, der autorisiert ist, darf im Mehrbenutzerbetrieb auf die gespeicherten Daten zugreifen.

1. **Flexibilität** zeigen

Die Daten müssen in beliebiger Form verknüpfbar sein.

1. **Effizient** sein

Die Zeiten für die Abfrage und für die Verarbeitung müssen kurz sein, ebenso für Änderungen und Ergänzungen des Datenbestandes.

1. **Datenschutz** gewährleisten

Die Daten sind vor unbefugtem Zugriff zu schützen.

1. **Datensicherheit** herstellen

Die Daten müssen gegen Programmfehler und Hardwareausfälle gesichert sein. Das Datenbanksystem soll nach Störungsfällen den korrekten Zustand wiederherstellen (recovery).

Die genannten Anforderungen sind idealtypisch und stehen miteinander in Konkurrenz. Weniger Redundanz wird z. B. mit geringerer Flexibilität und Effizienz erkauft.

1. Adams, R. (2019). SQL – Der Grundkurs für Ausbildung und Praxis (3. akt. Auflage). Hanser. [↑](#footnote-ref-2)
2. Fuchs, E. (2021). SQL – Grundlagen und Datenbankdesign. Herdt. [↑](#footnote-ref-3)