Praxisprojekt 4 (4. Semester)

an der

Berufsakademie Sachsen

Staatliche Studienakademie Riesa

Studiengang Labor- und Verfahrenstechnik

Kurs: RT14LV1 Studienrichtung: Biotechnologie

Thema: Konzeption und Entwicklung einer Browser-basierten Datenbank für das firmeninterne Chemikalienmanagement

Eingereicht von: Firma:

Martin Schneider quo data

Gesellschaft für

Qualitätsmanagement und Statistik mbH

Prellerstraße 14

01187 Dresden

Betrieblicher Betreuer: ............................

# Zusammenfassung

# Inhaltsverzeichnis

[1. Einleitung 1](#_Toc432420139)

[1.1 LIMS 1](#_Toc432420140)

[1.2 Webservices 1](#_Toc432420140)

[1.2.1 MySQL/Datenbanken 1](#_Toc432420143)

[1.2.2 PHP/HTML/CSS 1](#_Toc432420142)

[1.1.3 Vorteile/Nachteile 1](#_Toc432420143)

[1.3 Alternativen 2](#_Toc432420144)

[1.3.1 Webservices 1](#_Toc432420143)

[1.3.2 Client-Anwendungen 1](#_Toc432420143)

[2. Zielsetzung 5](#_Toc432420145)

[3. Software 6](#_Toc432420146)

[4. Planung 7](#_Toc432420150)

[4.1 Datenbank-Struktur 7](#_Toc432420151)

[4.2 Web-Interface 7](#_Toc432420152)

[4.3 Nutzerverwaltung 7](#_Toc432420153)

[4.4 Implementierung/Erweiterungen (Tools) 8](#_Toc432420154)

[5. Umsetzung 10](#_Toc432420155)

[5.1 Datenbank-Struktur 10](#_Toc432420156)

[5.2 Web-Interface 16](#_Toc432420158)

[5.3 Nutzerverwaltung 16](#_Toc432420158)

[5.4 Implementierung/Erweiterungen 16](#_Toc432420158)

[5.5 Dokumentation/Testprozeduren 16](#_Toc432420158)

[7. Zusammenfassung 22](#_Toc432420163)

[8. Literatur 24](#_Toc432420164)

# Tabellenverzeichnis

# Abbildungsverzeichnis

# Abkürzungsverzeichnis

HTML

PHP

CSS

W3C

DB

DBMS

# Einleitung

## LIMS

## Webservices

Laut W3C ist ein Webservice eine Softwareanwendung zur Machine-zu-Machine-Interaktion, die innerhalb eines Netzwerkes bereitgestellt wird. Die Beschreibung des Interfaces erfolgt in einem machinenlesbaren Format (Standard: WSDL). Die Kommunikation erfolgt typischerweise über HTML oder ähnliche Web-Standards. (Quelle: <https://www.w3.org/TR/ws-arch/#whatis>)

Ein Webservice ist typischerweise nach dem Server-Client-Prinzip aufgebaut, mit einem Server und einer Client-Anwendung. Typische Anwendungen für Webservices sind Synchronisationsdienste, Feedreader oder Datenbanksysteme.

Webanwendungen sind eine Unterkategorie der Webservices, die ein Interface für die Kommunikation mit Benutzern bereitstellen. Webanwendungen werden anders als normale Software nicht auf dem Rechner des Benutzers installiert, sondern laufen auf dem Server und werden typischerweise vom Benutzer mit einem Browser aufgerufen. Nutzereingaben werden auf dem Server verarbeitet und das Ergebnis an den Benutzer übergeben. Typische Anwendungen sind zum Beispiel Verbindungsauskünfte oder Flashgames. Es ist möglich, Teile der Logik und damit der Rechenlast der Webanwendungen auf den Client, also den Rechner des Benutzers, zu übertragen. Dieses Prinzip wird als „Fat Client“-Prinzip bezeichnet, im Gegensatz zum „Thin Client“-Prinzip, bei dem der Client nur die Darstellung der Ergebnisse übernimmt. Für einige Webanwendungen werden bestimmte Laufzeitumgebungen für den Browser benötigt, wie JavaScript oder Flash.

### Datenbanken/MySQL

Datenbanken sind eine Form der organisierten elektronischen Datenspeicherung mit dem Ziel, die Speicherung, Formatierung, Verwaltung und Manipulation der Daten konsistent, platzsparend und performant zu realisieren. Die Daten können abhängig von der Form der Datenbank Texte, Zahlen, logische Werte oder Dateien sein.

Datenbanken bestehen aus der Menge zu verwaltender Daten, der eigentlichen Datenbank, und dem Datenbankmanagementsystem (DBMS), das die Strukturierung und Verarbeitung der Daten realisiert. Um Zugriffe auf die Datenbank zu ermöglichen, bietet das DBMS eine Datenbanksprache an.

Anhand der vom DBMS vorgegebenen Strukturierung der Daten wird in verschiedene Datenbankmodelle eingeteilt, die die Beziehungen der gespeicherten Objekte zueinander angibt. Die wichtigsten Modelle sind:

* Hierarchisch
* Netzwerkartig
* Dokumentorientiert
* Objektorientiert
* Relational

Relationale Datenbanken bestehen aus mehreren Tabellen, in denen die Daten zeilenweise gespeichert werden. Die Datensätze verschiedener Tabellen können beliebig miteinander verknüpft werden. Die Verknüpfungen werden ebenfalls in Tabellen gespeichert. Der Aufbau einer Tabelle ist in Abbildung 1 anhand einer fiktiven Tabelle dargestellt.



Abbildung 1: Beispielhafter Aufbau einer Tabelle

Jede Tabelle besteht aus mehreren Attributen, die in Spalten dargestellt sind. Jede Zeile ist ein Datensatz oder „Tupel“, in dem ein Wert für jedes Attribut der Tabelle festgelegt ist. Das Tabellenschema gibt an, wie viele Attribute eine Tabelle hat sowie deren Namen und eventuelle Einschränkungen für die Werte, wie vorgeschriebene Datentypen, Einmaligkeit der Werte oder das Verbot von fehlenden Werten.

Jeder Datensatz muss über einen oder mehrere Schlüssel („keys“) eindeutig identifizierbar sein. In den meisten Fällen wird aus Gründen der Einfachheit und Skalierbarkeit ein fortlaufender Integer-Wert als ID genutzt, die dem Datensatz bei der Speicherung zugewiesen wird. Der Schlüssel bezieht sich nur auf den Datensatz, nicht auf die Position des Datensatzes in der Tabelle.

Sollen in einer relationalen Datenbank komplexere Daten verwaltet werden, können Tabellen untereinander verknüpft werden. Die Verknüpfung der Tabelle „Autos“ mit der Tabelle „Einwohner“ ist in Abbildung 2 dargestellt.



Abbildung 2: Verknüpfung von zwei Tabellen über einen foreign key

Um den Inhaber eines Autos in der Tabelle zu identifizieren, ist der Schlüssel des Datensatzes aus der Tabelle „Einwohner“ angegeben. Sollte ein Einwohner mehrere Autos besitzen, kann mehrfach auf den Eintrag verwiesen werden. Dadurch müssen die Daten des Einwohners nur ein einziges Mal angegeben werden und das Risiko für Inkonsistenzen zum Beispiel durch Schreibfehler wird vermieden. Ebenfalls vereinfacht wird die Manipulation der Daten. Wenn sich zum Beispiel die Adresse des Einwohners ändert, muss nur der entsprechende Datensatz aktualisiert werden. Die Aufspaltung von Tabellen zur Vermeidung von Redundanzen wird Normalisierung genannt. Eine vollständig normalisierte Datenbank enthält keine vermeidbaren Redundanzen mehr.

Die meisten relationalen Datenbanksysteme unterstützen als Datenbanksprache SQL (Structured Query Language), eine in den 70er Jahren entwickelte und international standardisierte Sprache zur Definition der Datenstruktur und Manipulation der Datenbestände relationaler Datenbanken. Vorteile von SQL sind die einfache Syntax und die an die englische Sprache angelehnte Semantik. Die Implementierung der Sprache ist Sache des DBMS, Modifizierungen zur Anpassung des Funktionsumfanges und der Performance sind gängige Praxis. Die systemspezifischen Implementierungen von SQL werden als Dialekte bezeichnet.

Ein sehr verbreitetes DBMS ist MySQL von Oracle. MySQL ist sowohl als Open-Source-Software als auch als kommerzielle Enterpriseversion erhältlich und ist auf allen verbreiteten Betriebssystemen lauffähig. Als Werkzeug zur Administration von MySQL-Datenbanken wird in den meisten Fällen die Webanwendung phpMyAdmin verwendet, die neben einem SQL-Terminal auch eine graphische Oberfläche zur Anzeige und Manipulation der Daten bietet. In Abbildung 3 ist ein Teil des Nutzerinterfaces von phpMyAdmin abgebildet, mit dem SQL-Terminal im oberen Bereich und der Anzeige von Datensätzen im unteren Bereich.

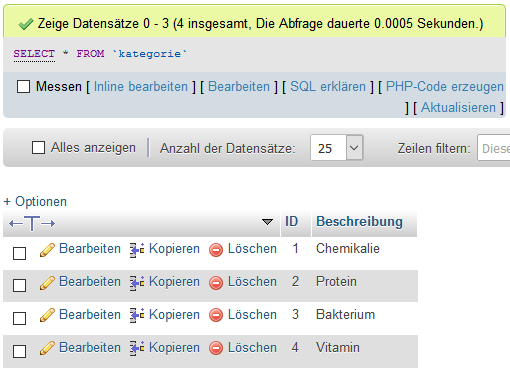


Abbildung 3: Ausschnitt aus der Oberfläche von phpMyAdmin

### PHP/HTML/CSS

PHP, HTML und CSS sind die derzeit führenden Webstandards für Webdesign und Webanwendungen. HTML (Hypertext Markup Language) ist eine Auszeichnungssprache, die für die Strukturierung von digitalen Dokumenten und die Ausstattung mit Inhalten wie Hyperlinks oder Bildern verwendet wird. HTML-Dateien werden mit Webbrowsern dargestellt und bilden die Grundlage des World Wide Web. Eine HTML-Datei enthält neben dem im Webbrowser angezeigten Text Angaben zur Formatierung und Metadaten wie die im Text verwendeten Sprachen oder den Namen des Autors. HTML unterstützt als Auszeichnungssprache keine Interaktivität. Um Websites interaktiv oder dynamisch zu gestalten, müssen Programmiersprachen wie PHP oder Javascript verwendet werden.

PHP (Hypertext Preprocessor) ist eine Skriptsprache, die zur Gestaltung dynamischer Websiten verwendet wird. Wird vom Client eine Anfrage an den Server gestellt, wird serverseitig aus dem PHP-Code dynamisch ein HTML-Dokument generiert und an den Client gesendet. Die Funktionsweise eines PHP-Skriptes ist in Abbildung 3 dargestellt.



Abbildung 4: Funktionsweise von PHP

Quelle: (https://www.neuhold.pro/Media/Default/Kurs%20PHP%20Beginner/PHP\_funktionsweise.png)

Der Server reagiert automatisch auf alle in der Anfrage erhaltenen Eingaben und erzeugt eine entsprechende HTML-Datei (seltener auch PDF- oder Bilddateien). Nachteil der serverseitigen Verarbeitung der Eingaben ist die hohe Rechenlast und Rechendauer für den Server, da bei jeder neuen Anfrage der Code neu verarbeitet werden muss.

CSS (Cascading Style Sheet) ist eine Stylesheet-Sprache, die für die Formatierung der HTML-Dokumente verwendet wird. Die Sprache wurde vom World Wide Web Konsortium entwickelt und wird beständig erweitert. Ziel der Sprache war eine Trennung von Inhalt und Design einer Website, um Designs seitenübergreifend und unabhängig vom Seiteninhalt wiederverwenden zu können. CSS kann für die Erstellung von externen Stylesheets (engl. für Formatvorlage) für die Seitenübergreifende Formatierung oder von internen Styles zur Formatierung von Textabschnitten innerhalb einer Datei verwendet werden. In den Stylesheets werden die mit HTML strukturierten Segmente der Datei mit Formatierungs-Attributen versehen, die der Browser interpretiert und bei der Anzeige der Datei berücksichtigt. In den Stylesheets sind Angaben zur Positionierung von Seitenelementen, zur Textformatierung und zur Formatierung des Browserfensters gespeichert. Die Interpretation der Angaben ist teilweise vom Browser abhängig, wodurch Seiten in verschiedenen Browsern unterschiedliche Designs haben können.

## Alternativen

### Eigensoftware

Eine Alternative zu der Webanwendung wäre ein Webservice mit einer entsprechenden Client-Software. Diese Software könnte auf jeder Sprache mit Bibliotheken für Datenbankanbindung geschrieben werden und würde für Datenbankzugriffe auf den Webservice des Servers zurückgreifen. Ein Vorteil dieses Ansatzes wäre eine deutlich geringere Serverlast, da nur SQL-Anfragen bearbeitet und keine HTML-Dateien generiert werden müssten. Auch die Vereinheitlichung des Layouts wäre einfacher, ebenso die Auswertung und Anzeige von Datensätzen. Die Wartezeiten beim Laden einer neuen Seite würden entfallen und die Funktionalitäten könnten kompakter zusammengeführt werden. Angriffe durch SQL Injections, also das Einbringen von schädlichem Code durch Benutzereingaben, könnten in einer Client-Anwendung leicht verhindert werden. Allerdings müsste die Kommunikation mit der Datenbank zusätzlich gesichert werden, um Zugriffe von Unbefugten zu verhindern. Ein weiterer Nachteil wäre die Plattformabhängigkeit der Client-Anwendung. Eine Webanwendung kann unabhängig vom Betriebssystem benutzt werden, eine Clientsoftware müsste für jedes System neu geschrieben werden. Weiterhin müsste auf jedem Rechner, der Zugriff auf die Datenbank benötigt, die Clientsoftware installiert werden.

Ein weiterer Ansatz wäre eine lokale SQLite-Datenbank mit einem Nutzer-Interface. SQLite-Datenbanken sind speziell für die Einbindung in Programme konfiguriert und nicht für den Zugriff über ein Netzwerk. Die direkte Einbindung der Datenbank bedingt also, dass Daten nicht auf andere Rechner synchronisiert werden. Es gäbe einen Rechner, auf dem das Programm installiert ist und auf dem die Datenbank gespeichert wird. Der Zugriff auf die Daten könnte nur über diesen Rechner erfolgen. Dieser Ansatz würde ein Maximum an Sicherheit garantieren, da eventuelle Angreifer Zugriff auf den Rechner mit der Anwendung benötigen würden. Allerdings wäre die Nutzung unkomfortabel und die Überprüfung von Beständen mit einem anderen Rechner nichtmöglich. Im Falle eines Festplattenschadens wäre außerdem die gesamte Datenbank verloren.

### Fremdsoftware

Es gibt bereits Softwarelösungen für die Inventarisierung von Chemikalien. Eine Software zur lokalen Speicherung und Bearbeitung ist das Programm „Chemikalienverzeichnis 6.6“, das eine Bibliothek vorgespeicherter Chemikalien und Gefahrstoffverordnungen mitliefert. Im Netzwerkbetrieb können Benutzer an anderen Rechnern über den Browser auf die lokal gespeicherte Datenbank zugreifen, auch ein Read-Only-Zugriff ist möglich. Der Preis für die Software liegt bei 419 € zzgl. Mehrwertsteuer.

Als kostenlose Alternative gibt es die Webanwendung „Quartzy.com“, die das Einrichten mehrerer Labore und Arbeitsgruppen mit jeweils getrennten Verzeichnissen erlaubt. Ebenfalls integriert ist eine Importfunktion über eine Excel-Tabelle und ein einfaches System für das Erstellen von Bestellanfragen. Es ist eine Kommunikationsfunktion für Mitglieder der gleichen Gruppe und eine Newsfeed-Funktion für Laboratorien integriert. Alle Daten werden auf einem Quartzy-Server gespeichert, der Zugriff ist nur über die Benutzeroberfläche möglich.

# Zielstellung

Ziel der Praxisarbeit war die Entwicklung und Implementierung eines browser-basierten Datenbanksystems zur firmeninternen Chemikalienverwaltung. Die Datenbank sollte auf einem internen Server liegen und über ein Webinterface für alle Rechner im Firmennetz erreichbar sein. Die Anforderungen an das Interface waren:

* Intuitiver, benutzerfreundlicher Aufbau
* Anpassung an das Firmen-Design
* Passwortschutz
* Datenmanipulation ohne manuelle Eingabe von SQL-Befehlen
* Bereitstellung von Grundfunktionen:
  + Einträge einfügen
  + Einträge bearbeiten
  + Einträge anzeigen, suchen oder filtern

Als Datenbank sollte eine mindestens in der ersten Normalform vorliegende relationale Datenbank mit dem Datenbankmanagementsystem MySQL verwendet werden. Die Datenbank sollte aus dem aktuellen Verzeichnis, einer Excel-Datei, heraus entwickelt werden.

# Software

Im Zuge dieser Arbeit wurde verschiedene Entwicklungsumgebungen für die Programmierung verwendet. Die verwendeten Entwicklungsumgebungen und Editoren waren:

* Netbeans DIE
* Microsoft Visual Studio C# 2010 Express
* Atom Editor
* Notepad++

Für die Strukturierung und Verwaltung der Datenbank wurde folgende Software verwendet:

* phpMyAdmin
* DB Designer Fork
* MySQL

Für die Einrichtung und den Betrieb des lokalen Webservers wurde das XAMPP-Paket genutzt.

Weiterhin wurde für die Dokumentation Microsoft Office 2010 und als Browser Mozilla Firefox eingesetzt.

# Theoretische Implementierung

## Datenbankstruktur

Das bestehende Verzeichnis in Form einer Excel-Datei enthielt alle gelieferten Chemikalien, mit jeder Lieferung als eigene Zeile. Zu jeder Lieferung wurden die folgenden Attribute gespeichert:

* Name der Chemikalie
* CAS-Nummer
* Kategorie
* Unterkategorie
* Lagerungsvorschrift
* Hersteller
* Menge
* Qualität/Spezifikationen
* Batch-/Chargennummer
* Summenformel
* Molmasse
* Gefahrenstoff ja/nein
* Zugehöriges Sicherheitsdatenblatt
* R-/S-Sätze
* H-/P-Sätze

Nach Normalisierung auf die 3. Normalform wurde das folgende Datenbankschema entwickelt:

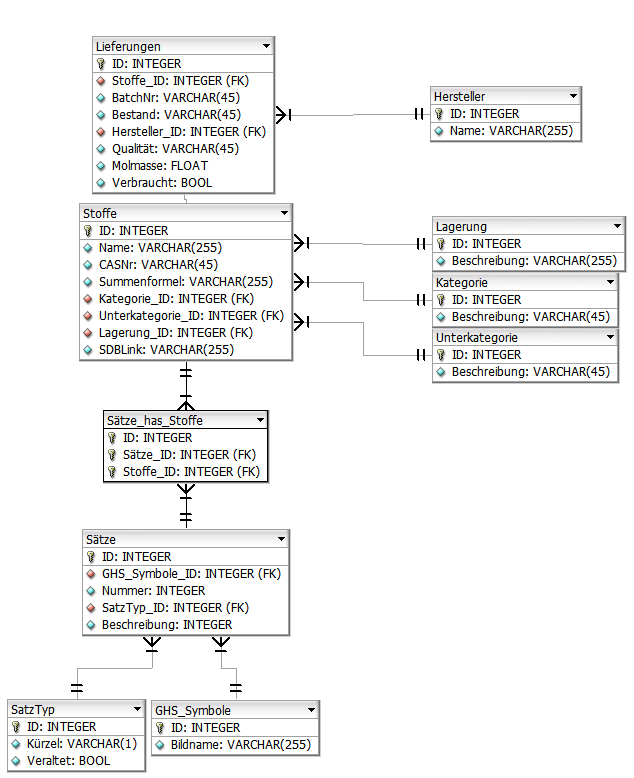


Abbildung : Datenbankschema abgeleitet aus dem bestehenden Excel-Verzeichnis, mit Tabellennamen, Attributen und zugehörigen Datentypen

Die vorgegebenen Attribute wurden ihrer Abhängigkeit nach in eigene Tabellen ausgelagert oder Tabellen zugewiesen. Hinter den Attributen ist in Klammern der geforderte Dateityp des Attributes angegeben, zusammen mit der maximalen Länge der Werte. Alle Stoffe wurden mit den stoffspezifischen Attributen in einer Tabelle gespeichert, die in den Lieferungen lediglich referenziert wird. Dadurch werden Redundanzen vermieden und Speicherplatz gespart. Auch die Lagerungsvorschriften, Kategorien, Unterkategorien und Hersteller wurden als eigene Tabellen umgesetzt.

Die in der Excel-Datei als Nummern gespeicherten Sätze wurden erweitert zu eigenen Datensätzen mit der zugehörigen Nummer, dem Satztyp, der Beschreibung und gegebenenfalls einem GHS-Symbol. Die Verknüpfung von Stoffen und Gefahrstoffsätzen ist eine m-zu-n-Beziehung, die über eine Tabelle mit gepaarten ID’s realisiert wurde.

## Web-Interface

## Nutzerverwaltung

# Zusammenfassung

# Literaturverzeichnis

<http://www.itwissen.info/definition/lexikon/Datenbank-DB-database.html>

<https://de.wikipedia.org/wiki/Datenbank>

<https://de.wikipedia.org/wiki/Relationale_Datenbank>

<https://de.wikipedia.org/wiki/Normalisierung_(Datenbank)>

<https://www.hdm-stuttgart.de/~riekert/lehre/db-kelz/chap4.htm> (Normalisierung)

<http://news.netcraft.com/archives/2013/01/31/php-just-grows-grows.html>

<https://w3techs.com/technologies/overview/programming_language/all>

<https://w3techs.com/technologies/history_overview/programming_language/ms/y>

<https://www.neuhold.pro/php/kapitel0>

<http://www.softguide.de/programm/chemikalienverzeichnis>

<http://chemikalienverzeichnis.de/>

E r k l ä r u n g

gemäß § 19 (Abs. 1) der Prüfungsordnung für den

Studienganges Labor- und Verfahrenstechnik vom 01. Oktober 2008

Ich habe die vorliegende Praxisarbeit selbst-

ständig verfasst und keine anderen als die an-

gegebenen Quellen und Hilfsmittel benutzt.

.................................... ............................... ...................................

(Ort) (Datum) (Unterschrift)