

Script - Modul 104

Christian Inauen

2019-01-10

1 Einführung

Dieses Script dient als Leitfaden für das Modul 104, Datenmodell implementieren, durchgeführt am BZT Frauenfeld.

2 Einführung ins Thema

Wir beginnen als Einstieg mit einem kurzen Negativbeispiel. Sie sollen anhand dieses Beispiels erkennen, dass ein korrekter und systematischer Umgang mit Daten für unsere Welt unabdingbar ist. Fehler können verheerende Folgen haben!

2.1 Lernziele

Jede/r Lernende:

- ...kann in eigenen Worten beschreiben, weshalb Datensätze in “einzelne Gruppen” aufgeteilt werden müssen
- ...kann Daten in in ihre Datentypen unterteilen (Zahlen, Buchstabenketten, Binärdaten)

2.2 Aufträge

Betrachten wir folgendes (Negativ-)Beispiel:

Name	Anschrift	Wohnort	Rechnungsnummer und -datum		Artikel	Rechnungsbetrag
Gustav Meier-Senn	Im Tobel 13	9248 Bichwil	Nr 001	01.02.2004	5, 8, 8, 8, 4, 9	SFr. 358.00
Ida Kuhler	Walchenstr. 45b	5224 Unterbözberg	Nr 002	01.02.2004	7,7,7	SFr. 90.00
Gustav Meier-Senn	Im Tobel 13	9248 Bichwil	Nr 004	20.03.2004	7,5,2,5,9	SFr. 750.00
Sylvia Brendel	Rennweg 7	8000 Zürich				
Gustav Meier-Senn	Im Tobel 13	9248 Bichwil	Nr 007	21.03.2004	8, 4, 5	SFr. 320.00
Morani	Birkenweg 16	1265 La Cure	Nr 008	22.03.2004	4, 8	SFr. 40.00
Sylvia Benz	Im Tobel 13	5600 Baden				
Walter Morani	Birkenweg 16	1265 La Cure	Nr 009	22.03.2004	9,1	SFr. 42.00
Merz, Beat	Seehalde 34	5600 Lenzburg	Nr 010	25.03.2004	6, 6, 9,8	SFr. 520.00

Figure 2.1: Negativbeispiel

Was passiert nun, wenn...

- sie die Adresse von Gustav Meier-Senn auf “Im Tobel 18” korrigieren*
- Walter Morani aus La Cure per heutigem Datum eine neue Rechnung bekommt, weil er die Artikel 4, 6, 2 und 9 gekauft hat
- Ida Kuhler alle Artikel der Rechnung Nr. 002 zurückgegeben hat

Notieren Sie sich die Resultate Ihrer Überlegungen.

2.3 Konsequenzen

Wir haben aus diesem Beispiel verschiedene Konsequenzen:

- - Für jeden Kunden eine eigene Tabelle erstellen
- - nur ein Artikel pro Feld, falls eine Rückgabe stattfindet
- kann man die Zeile einfach löschen.



3 Theoretische Hintergründe

In diesem Kapitel werden theoretische Hintergründe beleuchtet und erklärt. Die Inhalte wurden aus dem Compendio-Buch für dieses Modul übernommen (so mussten Sie es nicht kaufen).
Nachteil: Die Nummerierung der Überschriften stimmt nicht ganz.

3.1 Lernziele

Jede/r Lernende:

- ...kann mind. 3 Produkte für ein Datenbankmanagementsystem (DBMS) nennen
- ...kann in eigenen Worten beschreiben was SQL ist
- ...kann in eigenen Worten beschreiben was ein *Datenmodell* ist
- ...kann die vier Phasen der Datenbankentwicklung benennen

3.2 Einleitung

Einleitung

Informationen sind für unsere Wirtschaft sehr wertvoll. Für die Konkurrenzfähigkeit und das Überleben einer Firma ist es sehr wichtig,

- die für ihre Tätigkeit notwendigen Informationen zu besitzen,
- korrekte und aktuelle Informationen zu haben,
- auf diese Informationen jederzeit schnell und flexibel zugreifen zu können und
- die gespeicherten Informationen auswerten zu können.

Um diese Voraussetzungen zu erfüllen, muss eine genaue Analyse des Informationsbedarfs durchgeführt werden und müssen die Daten in einem System gespeichert werden, das die wirksame Verwaltung und das zuverlässige Abrufen oder Auswerten der Daten erlaubt. Dazu braucht es eine Datenbank.

Im ersten Teil des Lehrmittels lernen Sie fundamentale Begriffe rund um eine Datenbank kennen und erfahren mehr über das typische Vorgehen bei der Datenbankentwicklung. Zudem werden einige Konzepte und Tools vorgestellt, die bei der Entwicklung einer Datenbank für unsere Zwecke von grundlegender Bedeutung sind.

Figure 3.1: Einleitung

3.3 Was ist eine Datenbank

1.1 Datenbank, DBMS, SQL und Datenmodell

In einer **Datenbank** werden Daten abgelegt und in einer vorgegebenen Form (strukturiert) gespeichert. Für einen Anwender macht das alleinige Speichern von Daten aber keinen Sinn. Er möchte auch jederzeit die Möglichkeit haben, bestimmte Daten bzw. Informationen aus einer Datenbank «abzurufen» und auszuwerten.

Ein **Datenbankmanagementsystem (DBMS)** ermöglicht die effiziente Erstellung und Verwaltung bzw. Pflege einer Datenbank. Folgende DBMS sind weitverbreitet im Einsatz:

- Microsoft Access™
- Microsoft SQL™
- Oracle™
- DB2™
- MySQL™

Die meisten Datenbankmanagementsysteme verfügen über eine strukturierte Abfragesprache (**SQL**^[1]). Mit dieser Sprache kann auf der einen Seite die Struktur einer Datenbank erstellt und bearbeitet werden. Auf der anderen Seite können die gespeicherten Daten abgerufen und manipuliert werden. In diesem Lehrmittel werden Sie mit dem Datenbankmanagementsystem **MySQL** arbeiten. Dieses System ist sowohl für PCs unter Windows als auch für Linux-Maschinen verfügbar und kann darüber hinaus kostenlos vom Internet heruntergeladen werden.

Bevor Sie mit einem Datenbankmanagementsystem eine Datenbank erstellen können, müssen Sie die genauen Bedürfnisse in Erfahrung bringen und analysieren. Zu diesem Zweck ermitteln Sie zusammen mit den zukünftigen Benutzern die für das Informationssystem benötigten Daten und deren Beziehungen untereinander. Die Erkenntnisse der Analyse werden in einem Plan, dem sog. **Datenmodell**, visualisiert.

(#fig:02_wasistEineDatenbank)Was ist eine Datenbank

3.4 Entwicklung einer “Datenbank”

2 Typische Vorgehensweise

Eine Datenbank entsteht nicht einfach auf Knopfdruck. Damit Sie ein Datenmodell korrekt erstellen, implementieren und die Funktionen einer Datenbank wunschgemäß nutzen können, sind verschiedene Phasen der Datenbankentwicklung zu durchlaufen. In diesem Kapitel erfahren Sie, welche Hauptaufgaben dabei anstehen.

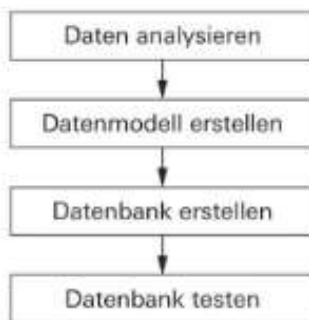
2.1 Phasen der Datenbankentwicklung

Der Weg zu einer fertigen Datenbanklösung lässt sich in folgende **Phasen** gliedern:

1. Daten analysieren
2. Datenmodell erstellen
3. Datenbank erstellen
4. Datenbank testen

Nachfolgend sehen Sie die einzelnen **Schritte** auf diesem Weg in einer Übersicht:

[2-1] Phasen der Datenbankentwicklung



(#fig:03_PhasenEntwicklung)Phasen der Entwicklung

4 Entitäten und Attribute

Gemäss Wikipedia ist eine *Entität*:

Als Entität (auch Informationsobjekt genannt, englisch entity) wird in der Datenmodellierung ein eindeutig zu bestimmendes Objekt bezeichnet, über das Informationen gespeichert oder verarbeitet werden sollen. Das Objekt kann materiell oder immateriell, konkret oder abstrakt sein. Beispiele: Ein Fahrzeug, ein Konto, eine Person, ein Zustand.

Dies bedarf einer Erklärung - dem wollen wir uns in diesem Kapitel annehmen.

4.1 Lernziele

Jede/r Lernende:

- ...kennt die Begriffe Entität, Tupel und Attribut und kann Sie sinngemäss wiedergeben.
- ...kann aus einer Beschreibung Entitäten selektieren
- ...kann zu Entitäten die Attribute hinzufügen
- ...kann Entitäten graphisch darstellen

4.2 Entität und Tupel

Eine *Entität* (philosophisch auch das *Ding*), ist ein Objekt, welches alle Eigenschaften eines *Dings* beinhaltet. Nicht mehr - aber eben auch nicht weniger.

Das Festlegen von Entitäten funktioniert meistens, in dem man alle *Substantive* in einer Aufgabenbeschreibung markiert.

Machen wir ein Beispiel:

In Büroräumen sitzen Mitarbeiter. In den Zimmern sind Telefone aufgestellt.

Dies führt man unserer Analyse zu folgenden Entitäten:

- Büroräume
- Mitarbeiter
- Telefone

In diesem Beispiel werden vor allem *physische* Objekte benannt. Sie können aber auch abstrakter sein...zum Beispiel:

- Ein Individuum wie beispielsweise ein Arzt, ein Patient, eine Krankenschwester usw.
- Ein reales Objekt wie beispielsweise ein Operationssaal, ein Krankenzimmer usw.
- Ein abstraktes Konzept wie beispielsweise eine Diagnose, ein Fachgebiet usw.
- Ein Ereignis wie beispielsweise ein Kreislaufkollaps, eine Patientenaufnahme usw.

Aus der Sicht des Modellentwerfers kann eine Entität folgendermassen charakterisiert werden:

- Eine eindeutig identifizierbare Einheit.
- Eine Einheit, deren Existenz auf einem geeigneten Speichermedium aufgrund eines Identifikationsmerkmal darstellbar sein muss (Schlüssel).
- Eine Einheit, für die Informationen zu sammeln und auf einem geeigneten Speichermedium festzuhalten sind.

Diese Merkmale sind bei der Ermittlung der ``Ankerpunkte`` eines Datenmodells von entscheidender Bedeutung.

4.3 Tupel

Ein *Datensatz* einer Entität (also "der Büroraum 207", "der Mitarbeiter YY") bezeichnet man als **Tupel**.

4.3.1 Entitätsmenge

Die Entität ist der Einzelfall. Bei Datenbanken fasst man eine Entität zusammen. Es gibt ja schliesslich nicht nur eine Bestellung, es gibt Bestellungen von vielen Kunden. Man spricht dann auch von einer Entitätsmenge. Als Überschrift wird dann oft einfach die Einzahl der Entitäten verwendet.

4.4 Attribute



Damit Entitäten beschrieben, und mit Inhalten gefüllt werden können, werden sie mit *Attributen* angereichert. Diese Eigenschaften können zu folgenden Dingen nützlich sein:

- Charakterisierung (Informationen)
- Klassierung (Gruppierungen)
- Identifizierung (das wären dann Schlüssel, kommt später)

Bleiben wir bei unserem Beispiel - und erweitern wir die Aufgabenbeschreibung:

In Büroräumen (charakterisiert durch eine Zimmernummer) sitzen seit einem Zeitpunkt Mitarbeiter (Personalnummer, Name, Titel, Status) an einem bestimmten Platz. In den Zimmern sind Telefone (besitzen eine eindeutige Telefonnummer) aufgestellt, die als Hausapparat oder Amtsapparat geschaltet sind.

Die führt zu folgenden Entitäten, mit den jeweiligen Attributen:

- Büroraum
 - Attribut: Zimmernummer
- Mitarbeiter
 - Attribut: Personalnummer
 - Attribut: Name
 - Attribut: Titel
 - Attribut: Status
- Telefon
 - Attribut: Telefonnummer
 - Attribut: IstHaustelefonOderNicht

Was jetzt natürlich noch fehlt, sind die Beziehungen zwischen den Entitäten.

4.4.1 Darstellung

Sie sehen schon bei der Auflistung der Inhalte oben, es wäre schon diese Inhalt graphisch darstellen zu können. Entitäten werden als Vierecke mit Attribut-Inhalten dargestellt - unser Beispiel sieht also wie folgt aus:

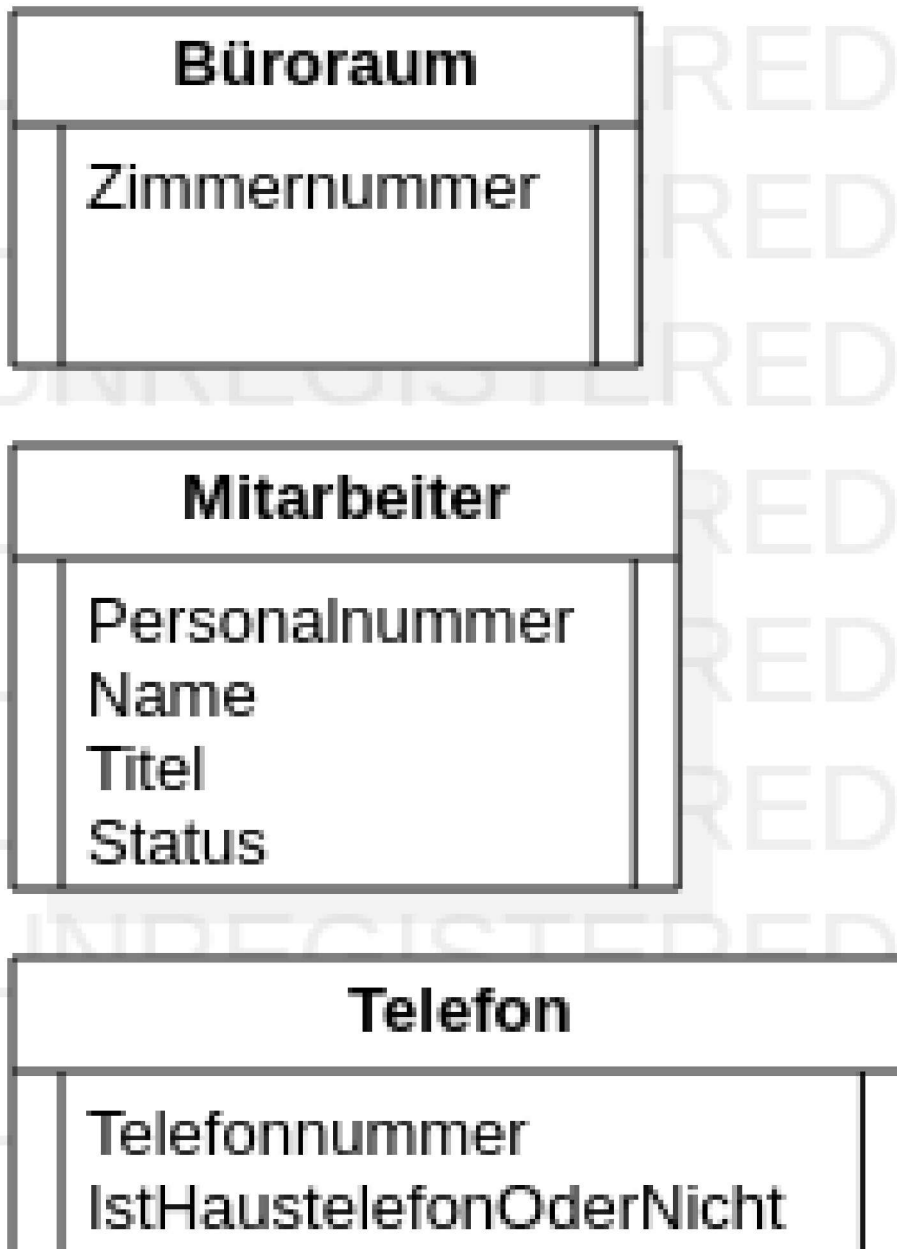


Figure 4.1: Ohne Relationen

4.5 Ausblick : Begriff Domänen

Eine Domäne beschreibt den zulässigen Wertebereich eines Attributs. Das können fest vorgegebene Werte sein (z.B. Januar, Februar, ...), Bereiche (z.B. von 0 bis 999, von A bis G) oder Mengenangaben bzw. Datentypangaben (z.B. natürliche Zahl, reelle Zahl, Datum).

Wir werden uns später genauer mit diesen Wertebereichen auseinandersetzen, da diese auch DBMS-Produkt-abhängig sein können.



5 Übungen : Entitäten und Attribute

5.1 Movies

Sie wollen ein Datenbanklösung bauen, die Informationen zu Filmen (oder meinetwegen: Netflix-Serien) speichern kann. Folgende Gegebenheiten sind zu berücksichtigen:

Filme werden von Regisseuren und Filmstudios produziert. Dabei werden Filme jeweils durch ihren Titel und das Produktionsjahr beschrieben. Regisseure haben Namen, Adressen und Telefonnummern. Zusätzlich werden sie über Emails kontaktiert. Die Filmstudios werden von mehreren Besitzern geführt. Die Studios haben einen Namen, eine Adresse und ein Gründungsjahr.

Auftrag:

Erstellen Sie auf Papier ein Diagramm in dem Sie die Entitäten mit Attributen darstellen. Die Beziehung zwischen den Entitäten spielen hier noch keine Rolle.

Digitalisieren Sie das Bild mit einem Foto und speichern Sie es auf OneNote

5.2 Baustellen

Sie wollen eine Datenbanklösung bauen, die Informationen zu Baustellen speichern kann. Folgende Gegebenheiten sind zu berücksichtigen:

Baustellen werden Bauherren geführt. Diese besitzen einen Namen, Adressen und Telefonnummern. Auf den Baustellen sind diverse Bagger (mit Baujahr, Anzahl Rädern, Modell und Hersteller) und Lastwagen (Baujahr, Anzahl Räder, Modell, Hersteller, KFZ-Nummern) im Einsatz. Jede nach Bau arbeiten auf der Baustelle verschiedene Handwerker, beschrieben durch Namen, Adressen und Telefonnummern. Die Handwerker wiederum gehören zu einem Unternehmen. Jedes Unternehmen besitzt einen Namen, einen Besitzer und eine Standortadresse. Die Unternehmen sind jeweils durch eine Haupttelefonnummer erreichbar.

Auftrag:

Erstellen Sie auf Papier ein Diagramm in dem Sie die Entitäten mit Attributen darstellen. Die Beziehung zwischen den Entitäten spielen hier noch keine Rolle.

Digitalisieren Sie das Bild mit einem Foto und speichern Sie es auf OneNote



6 Relationen, Kardinalitäten

Bei den vorhergehenden Übungen hat etwas Wichtiges gefehlt: Die Beziehungen zwischen Entitäten. Man nennt diese Beziehungen *Relationen*. Im folgenden Kapitel befassen wir uns damit, wie diese Relationen ausgebildet und beschrieben werden können.

6.1 Lernziele

Jede/r Lernende:

- ...kann den Begriff *Relation* in eigenen Worten beschreiben
- ...kann den Begriff *Kardinalität* in eigenen Worten beschreiben
- ...kann verschiedene Kardinalitäten in einem ER-Diagramm anwenden

6.2 Relationen

Wenn Entitäten in Beziehung zu einander stehen, spricht man davon, dass Entitäten in *Relation stehen*. Eine Relation zwischen zwei Entitäten wird durch eine verbundene Linie zwischen den Entitäten dargestellt.



Figure 6.1: Eine Relation

Diese Darstellung bedeutet:

Lehrer stehen (irgendwie) mit Schülern in Relation

6.3 Kardinalitäten / Beispiele und Einführung

Bei genauerem Hinschauen merken Sie, dass nicht alle Relationen “gleich” sind.

Beispiel 1:

Jeder Mensch besitzt eine Nase.

Folgende Fakten sind wohl wahr (wenn *Nase* und *Mensch* Entitäten sind):

- Jeder Mensch besitzt eine Nase (und zwar genau eine)
- Jede Nase gehört zu einem Menschen (und zwar genau zu einem Menschen)

Man nennt dies eine **1 zu 1 Relation**

Anders verhält es sich mit den Nasenhaaren...

Beispiel 2:

Mehrere Nasenhaare gehören zu einer Nase

Folgende Fakten sind wohl wahr (wenn *Nase* und *Nasenhaare* Entitäten sind):

- Nasenhaare (im Minimum wohl eines - als Maximum unzählige) gehören zu einer Nase (und zwar zu genau einer)
- Nasen besitzen Nasenhaare (im Minimum wohl eines - als Maximum unzählige)

Weil man jetzt diese Nasenhaare nicht zählen kann (oder will), nennt man die Anzahl einfach "n" (...eine normale mathematische Variable).

Man nennt dies eine **1 zu n Relation** ... wobei die Richtung der Relation von Bedeutung ist!

Die Darstellung sieht wie folgt aus:

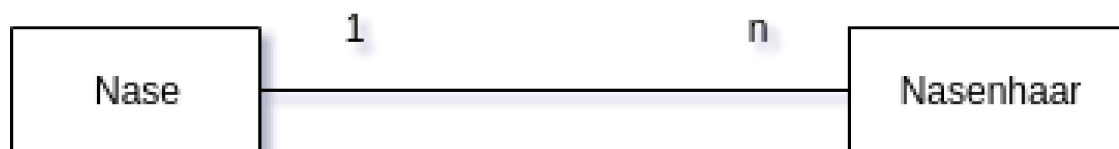


Figure 6.2: Nase zu Nasenhaare

Analog zum letzten Beispiel, können jetzt noch weitere Kombinationen aus Kardinalitäten möglich sein. Der folgende Abschnitt gibt eine Übersicht dazu.

6.4 Kardinalitäten / Theorie

Um Kardinalitäten einheitlich zu benennen, gibt es verschiedene Möglichkeiten. Die folgende Liste hat sich (für uns) bewährt:

Zehn- der	math. Bedeu- tung	umgangssprachlich
1	1	
c	0, 1	
m	1..m	
mc	0, 1..m	

Figure 6.3: Kardinalitäten

Auftrag:

Befüllen Sie die Tabelle mit umgangssprachlichen Bedeutungen für die verschiedenen Bezeichnungen

Die folgenden Bilder stellen Beispiele zu verschiedenen Relationen dar.



Figure 6.4: Kardinalitäten mit Erklärungen

Im folgenden eine Reihe von Beispielen:

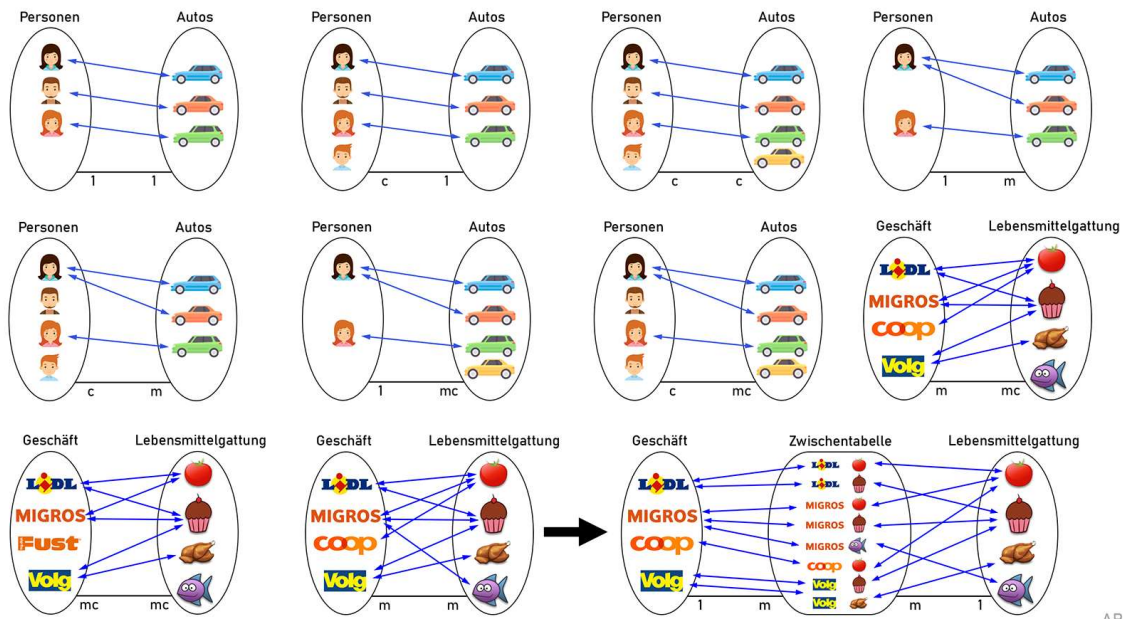


Figure 6.5: Beispiele zu Relationen

6.5 Kardinalitäten / Mathematischer Hintergrund

Natürlich lassen sich die Relationen mathematisch korrekt abbilden (das Ganze ist im Wesentlichen Teil der *Mengenlehre*). Die folgende Abbildung zeigt alle möglichen Kombinationen aus Kardinalitäten:

Tabelle2 \ Tabelle1	1	c	m	mc	
1	1-1	1-c	1-m	1-mc	← Hierarchische Beziehungen
c	c-1	c-c	c-m	c-mc	← Konditionelle Beziehungen
m	m-1	m-c	m-m	m-mc	← Netzwerkförmige Beziehungen
mc	mc-1	mc-c	mc-m	mc-mc	

Figure 6.6: Mathematische Abbildung der Kardinalitäten

7 Übungen : Relationen

7.1 Einstieg

Tragen Sie in den folgenden Relationen die "sinnvollen" Kardinalitäten ein:



Figure 7.1: Relationsaufgabe 1



Figure 7.2: Relationsaufgabe 2

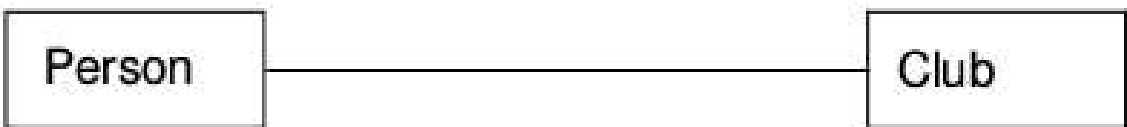


Figure 7.3: Relationsaufgabe 3

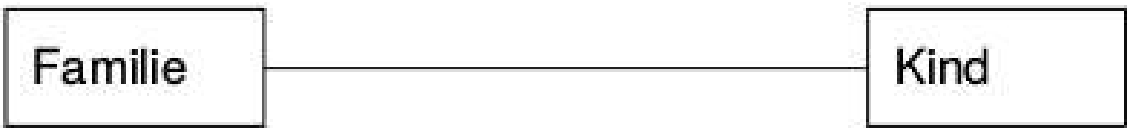


Figure 7.4: Relationsaufgabe 4



Figure 7.5: Relationsaufgabe 5

7.2 Liegenschaftsverwaltung

7.2.1 Entitäten

Eine Verwaltungsgesellschaft möchte ihre Liegenschaftsverwaltung automatisieren. Die Gesellschaft verwaltet die Liegenschaften verschiedener Eigentümer (eine Liegenschaft gehört jeweils nur einem Eigentümer) und vermietet diese weiter. In der Regel enthält eine Liegenschaft mehrere Mietobjekte, über die mit einem Mieter ein Mietvertrag abgeschlossen wird. Es kann jedoch auch vorkommen, dass bei Mieterwechsel gleichzeitig mehrere Mietverträge für ein Mietobjekt existieren.

Auftrag:

Tragen Sie in die Kästchen unten die Entitäten ein

The diagram consists of five empty rectangular boxes arranged in a scattered pattern, intended for the user to enter the names of the entities in the database.

Figure 7.6: Liegenschaftsverwaltung Teilaufgabe 1

7.2.2 Kardinalitäten

Auftrag:

Tragen Sie in die Kästen oben die Relationen ein. Benutzen Sie dazu die Werte *1, c, m*.

7.2.3 Attribute

Auftrag:

Überlegen Sie sich möglich Attribute zu den Entitäten. Sammeln Sie Ideen und notieren Sie sie.



8 Primärschlüssel, Fremdschlüssel und Index

Betrachten wir im Folgenden Ihre Situation in der Berufsschule und in Ihrem Lehrbetrieb. Wir konzentrieren uns dabei auf die Datenhaltung in unserer Schule.

Achtung: Die folgenden Inhalte beziehen sich ausschliesslich auf relationale Datenbanken (MySQL / MSSQL / Oracle usw.)!

8.1 Lernziele

Jede/r Lernende:

- ...kann in eigenen Worten beschreiben was ein Primärschlüssel ist
- ...kann in eigenen Worten beschreiben was ein Fremdschlüssel ist
- ...kann Primärschlüsselkandidaten aus einem Datensatz bestimmen
- ...kann 1 x mc-Relationen in einem ERM-Diagramm abbilden
- ...kann mc x mc-Relationen in einem ERM-Diagramm abbilden

8.2 Ausgangssituation

Sie sind als Lernende in unserer Schule angemeldet. Dabei wurden die Daten Namen, Adresse, Telefonnummer von Ihnen gespeichert. Zusätzlich wurden Sie mit einem Kürzel ausgestattet. Ihre Lehrmeister, mit Namen, Telefonnummer und Email, befinden sich ebenfalls in dieser Datenbank. Natürlich wurde auch die Verbindung von Ihnen mit Ihrem Lehrmeister gespeichert.

Daneben werden in der Datenbank Klassen administriert. Von Klassen wird grundsätzlich nur der Kürzel gespeichert. Ebenfalls werden Sie als Lernender einer Klasse zugeordnet.

8.3 ER-Diagramm

Die Ausgangslage führt zu folgendem ER-Diagramm:

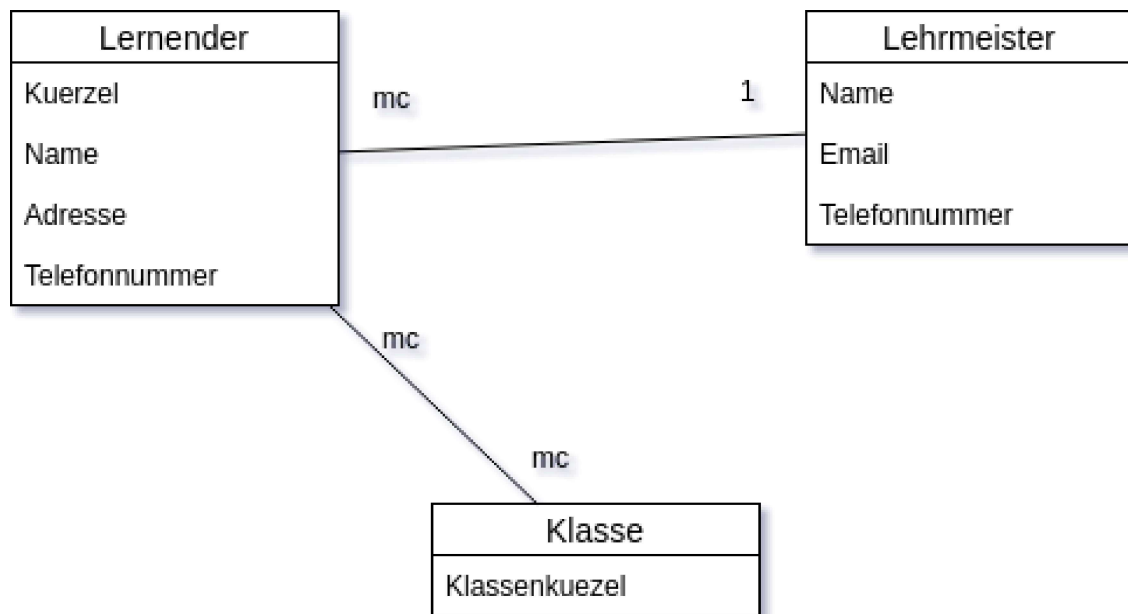


Figure 8.1: Ausgangslage

8.4 Primärschlüssel

PrimaryKey (PK):

Der Primärschlüssel ist eine Spalte in der Tabelle, durch die eindeutig jede Zeile identifiziert wird (gerne mit dem Namen ID). Es kann auch eine Kombination von Spalten als eindeutig festgelegt werden; das ist aber selten sinnvoll. In der Regel sollte diese Spalte auch keine andere "inhaltliche" Bedeutung haben als die ID.

Das Relationenmodell fordert, dass jede Tabelle ein Attribut oder eine Attributkombination enthält, über die jeder Datensatz eindeutig identifiziert werden kann. In der Praxis kommen Attribute, die in jedem Fall eindeutig sind, selten vor. Beispiele wären Autonummern oder eine Kombination aus Bankleitzahl und Kontonummer. Oft werden daher zusätzliche Attribute definiert, welche die Datensätze fortlaufend oder nach einem Nummernkreis durchnummerieren. Sie können vor dem Benutzer verborgen bleiben und nur der internen Datenbankverwaltung dienen oder sichtbar sein, wie zum Beispiel Artikel- oder Kundennummer.

Solche eindeutigen Felder werden auch auf der physikalischen Ebene benötigt und dienen unter anderem dem schnellen Auffinden von Datensätzen in grossen Tabellen. Sie werden Primärschlüssel genannt. Eine Tabelle kann nur einen Primärschlüssel haben. Die Tabelle

erscheint nach diesem Schlüssel sortiert. Ein Primärschlüssel ändert während der gesamten Existenz eines Datensatzes seinen Wert nicht.

Der Primärschlüssel wird speziell gekennzeichnet, z. B. fett, eine spez. Farbe (meist rot) oder doppelt unterstrichen. Es kann auch vorteilhaft sein bei der Benennung ein p voran zu stellen.

Beispiel: pKundenNR

Um jetzt also in unserem Klassenbeispiel die Primärschlüssel zu finden, gehen wir wie folgt vor:

Der Primärschlüssel muss den Datensatz eindeutig identifizieren

Folgende Primärschlüssel machen Sinn:

- Klasse -> Attribut *Klassenkuerzel*
- Lerndender -> Attribut *Kuerzel*
- Lehrmeister -> Attribut *Name*

8.5 Fremd-Schlüssel

ForeignKey (FK):

Über Fremdschlüssel werden die Tabellen miteinander verknüpft. Einem Feld in der einen Tabelle wird ein Datensatz in einer anderen Tabelle zugeordnet; dieser wird über den Primärschlüssel bereitgestellt. Es kann auch eine Kombination von Spalten verwendet werden; da sich der Fremdschlüssel aber auf einen Primärschlüssel der anderen Tabelle beziehen muss, ist dies ebenso selten sinnvoll. Die "Datenbank-Theorie" geht sogar soweit, dass die Schlüsselfelder dem Anwender gar nicht bekannt sein müssen.

Ein Fremdschlüssel ist ein Feld in einer Relation, welches eine Beziehung zu einem Schlüsselfeld einer anderen Relation herstellt. Beispielsweise befindet sich in einer Relation Aufträge in jedem Datensatz eine Kundennummer des Kunden, der den Auftrag ausgelöst hat.

Diese Kundennummer identifiziert einen Kunden in der Kundenrelation eindeutig. Die Kundennummer in der Auftragsrelation stellt somit einen Fremdschlüssel ("fremder" Schlüssel einer anderen Relation, wo er Primärschlüssel ist) dar.

Fremdschlüssel werden einfach unterstrichen gekennzeichnet, auch kursiv ist gebräuchlich! Empfehlenswert ist auch hier ein Präfix zu verwenden.

Fremdschlüssel dienen also dazu, "fremde" Primärschlüssel als Verbindung einzutragen.

Beispiel: fKundenNr

Ziel ist:

Die Relationen gemäss den Kardinalitäten abzubilden.

8.6 Relation -> 1 x mc abbilden (Lernender - Lehrmeister)

Die Interpretation dieser Relation bedeutet:

Es gibt offenbar eine Reihe von Lehrmeister, die mehrere oder keine Lernenden betreuen können. Jeder Lernende besitzt genau einen Lehrmeister.

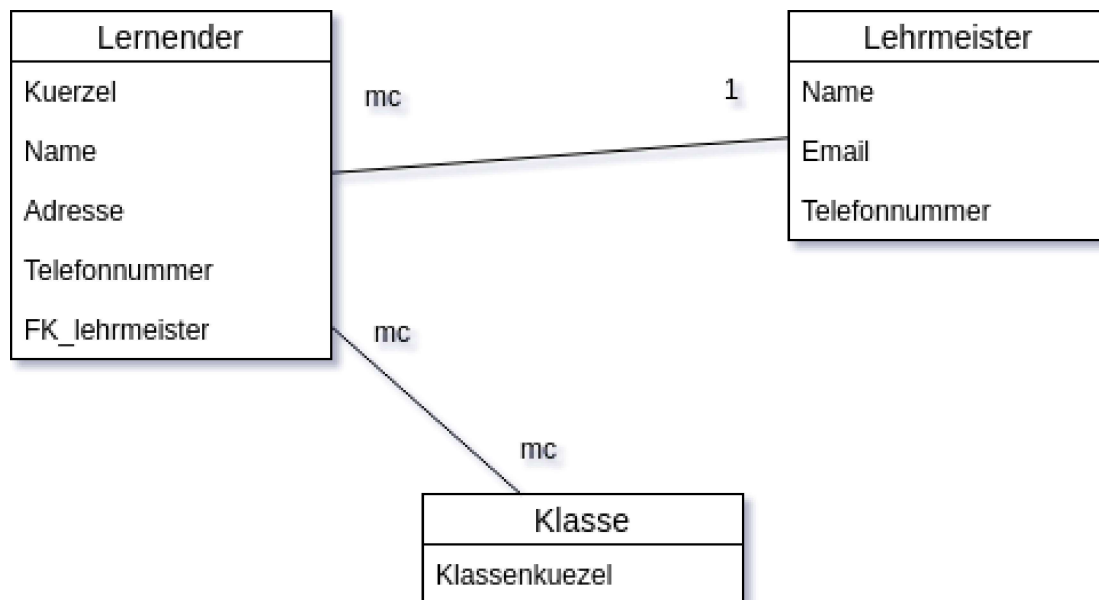
Um dieses Beispiel jetzt im Datenmodell abzubilden, wird auf der Entität "Lernenden" ein zusätzliches Attribut erfasst:

FK_lehrmeister

Der Typ des Attributs entspricht dem Primärschlüssel der Entität "Lehrmeister". Wurde also bei uns der Lehrmeister über den Namen identifiziert (was eigentlich nicht so clever ist), wird der Name als auch auf dem Lernenden als Fremdschlüsselbeziehung zum Lehrmeister eingetragen.

Folgendes Bild zeigt die Erweiterung mit dem Attribut FK_lehrmeister:





Text

Figure 8.2: Lernender - Lehrmeister

Damit wird jetzt also jedem Lernenden-Tupel noch zusätzlich mitgespeichert, wie der Name des Lehrmeisters lautet. Und da das für alle Lernenden möglich ist, können auch mehrere Lernenden den selben Lehrmeister besitzen.

Praktisch sieht das so aus:

A	B	C	D	E
Tabelle: Lernender				
Kuerzel	Name	Adresse	Telefonnummer	FK lehrmeister
21adsfdf	name1	adresse1	64544445	NameLM2
22adsfdf	name2	adresse2	64544446	NameLM4
23adsfdf	name3	adresse3	64544447	NameLM2
24adsfdf	name4	adresse4	64544448	NameLM4
25adsfdf	name5	adresse5	64544449	NameLM3
26adsfdf	name6	adresse6	64544450	NameLM6
Tabelle: Lehrmeister				
Name	Email	Telefonnummer		
NameLM1	1test@test.com	64644544		
NameLM2	2test@test.com	64644545		
NameLM3	3test@test.com	64644546		
NameLM4	4test@test.com	64644547		
NameLM5	5test@test.com	64644548		
NameLM6	6test@test.com	64644549		

Figure 8.3: Lernender - Lehrmeister mit Daten

Fragen:

- Welche Lernende gehören zu welchem Lehrmeister?
- Welcher Lehrmeister besitzt keine Lernenden?

8.6.1 Relation -> 1 x 1 abbilden

Die 1 x 1 Relation bildet hier nur ein Spezialfall der 1 x n Relation. Bei der 1 x 1 -Relation werden einfach auf beiden Entitäten ein FK-Attribut eingeführt.

8.7 Relation -> mc x mc abbilden (Klasse - Lernender)

Die 1 x mc-Beziehung war noch relativ einfach - jetzt wirds ein bisschen komplizierter. Wenn wir jetzt die Relation zwischen Lernenden und Klassen abbilden wollen, müssen wir beachten, dass Lernende grundsätzlich in mehreren Klassen sein können. Und eine Klasse besteht aus mehreren Lernenden.

Damit wird diese Relation abbilden können, müssen wir eine Zwischentabelle *RelationKlasse* einführen. In ihr werden jetzt alle Relationen als Kombination zwischen Primärschlüsseln der Entitäten abgebildet.

Aber langsam : zuerst das Bild.

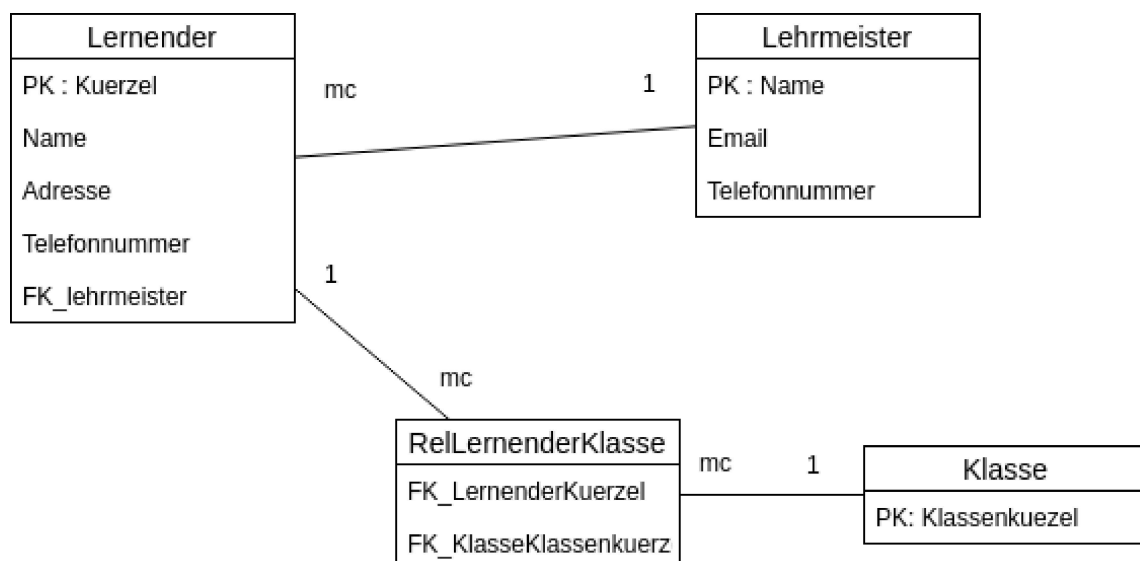


Figure 8.4: Klassen - Lernender

Auch das kann jetzt noch ein bisschen “hirnverknötend” sein...darum unser Beispiel:

	A	B	C	D	E
1	Tabelle: Lernender				
2	<u>Kuerzel</u>	<u>Name</u>	<u>Adresse</u>	<u>Telefonnumm</u>	<u>FK Lehrmeister</u>
3	21adsfdf	name1	adresse1	64544445	NameLM2
4	22adsfdf	name2	adresse2	64544446	NameLM4
5	23adsfdf	name3	adresse3	64544447	NameLM2
6	24adsfdf	name4	adresse4	64544448	NameLM4
7	25adsfdf	name5	adresse5	64544449	NameLM3
8	26adsfdf	name6	adresse6	64544450	NameLM6
9					
0					
1					
2	Tabelle: RelLernenderKlasse				
3	<u>FK LernenderKuerzel</u>	<u>FK KlasseKlassenkuerzel</u>			
4	21adsfdf	IN18-22a			
5	23adsfdf	IN18-22a			
6	25adsfdf	IN18-22a			
7	21adsfdf	TbmKlassea			
8	23adsfdf	TbmKlassea			
9	26adsfdf	TbmKlasseb			
0					
1					
2					
3	Tabelle: Klasse				
4	<u>Klassenkuerzel</u>				
5	IN18-22a				
6	IN18-22b				
7	TbmKlassea				
8	TbmKlasseb				
9					

Figure 8.5: Lernender - Lehrmeister mit Daten

Fragen:

- Welche Lernende sind in zwei, oder mehr Klassen?
- Welche Lernende sind nur in einer Klasse?
- An wie viele Klassenevents muss der Lehrmeister “NameLM2” gehen? Welche?

8.8 Abschluss und Feststellungen

Folgende Dinge gilt es festzuhalten:

Die anderen Kardinalitäten sind als Spezialformen der genannten 1 x mc-Relationen und mc x mc-Relationen zu sehen.

Sind in einem ER-Diagramm (Entitäten und Attribute) alle Zwischentabellen vorhanden und aufgelöst, spricht man von einem ERM-Diagramm (Entity Relationship Modell)

Die Überführung aus mc x mc-Relationen mittels Zwischentabellen, ist nur für relationale DBMS nötig.



9 MS-Access installieren

Das folgende Kapitel zeigt, wie Sie MS-Access auf einer Windows-10-VM der Schule installieren. Sie dürfen, falls Ihr Account es zulässt, die MS-Access-Installation auch lokal auf Ihrem Laptop ausführen.

9.1 Lernziele

Jede/r Lernende:

- ...hat eine funktionierende MS-Access-Installation zur Verfügung (Laptop oder VM)

9.2 Access Download

1. Loggen Sie sich auf Dreamspark mit Ihrem Account ein
2. Suchen Sie unter “Anwendungen” nach Access 2016
3. Laden Sie die Anwendung herunter (ist eine ISO-Datei) und merken Sie sich (<- machen Sie einen Printscreen) den Key

9.3 Access Installation

1. Starten Sie Ihre Windows-VM
2. Hängen Sie die ISO-Datei an Ihrer VM an
3. Starten Sie die Installation von der angehängten CD

9.4 Abschluss

Sie haben dieses Kapitel abgeschlossen, wenn:

- eine MS-Access-Installation auf Ihrem Laptop (oder in der VM) zur Verfügung steht



10 MS-Access - ein Einführung (Praxis)

Das folgende Kapitel bearbeitet anhand von einem Einführungsvideo die wichtigsten ersten Schritt mit MS-Access.

10.1 Lernziele

Jede/r Lernende:

- ...kann die Desktop-Applikation MS-Access öffnen
- ...kann eine leere Datenbank erstellen
- ...kann eine Tabelle einer Datenbank hinzufügen und Daten von Hand erfassen

10.2 Das Video

Betrachten Sie in Einzelarbeit das folgende Video um machen Sie die gemachten Schritte im Video selbst mit.

Video-URL: <https://www.youtube.com/watch?v=aivWWVRjATo>

Access: Einstieg - Öffnen, Oberfläche, erste Tabelle anlegen



10.3 (freiwilliges) Lernvideo

Das folgende Lernvideo zeigt Ihnen, zugegebenermassen etwas langatmig, die Idee von Access. Wer also Access noch nicht kennt: Hier gibt es eine Vorstellung der Möglichkeiten.

Video-URL: <https://www.youtube.com/watch?v=fWM8G-Xrcx4&list=PLDBHHU1FGOoNbL29LLxEI61SXUA42f7LV>



10.4 Lernzielkontrolle

Sie sollten nun in der Lage sein:

- Eine Datenbank zu erstellen
- Tabellen zu erfassen
- Daten in Tabellen zu erfassen

11 MS-Access - ein Einführung (Theorie)

Nachdem Sie im vorhergehenden Video schon einmal MS-Access in der Anwendung gesehen haben, möchten wir in folgendem Kapitel diverse “Access-Begriffe” und Konventionen einführen.

Wir arbeiten dabei mit mehreren Medien:

- Video-Tutorials von Youtube
- HTML-Tutorial von www.access-tutorial.de

Dieses Kapitel soll Ihnen als Leitfaden durch die Inhalte dienen - bearbeiten Sie also die Schritte aus diesem Kapitel durch und konsumieren Sie die anderen Medien nach Anleitung.

Das HTML-Tutorial von www.access-tutorial.de steht Ihnen als download-bare ZIP-Datei zur Verfügung (Moodle).

11.1 Lernziele

Jede/r Lernende:

- ...weiss was Access ist und wofür es verwendet werden kann
- ...kennt die Begriffe *Tabellen Abfragen Formulare Berichte Makros Module* und kann sie in eigenen Worten beschreiben
- ...kennt die Namenskonventionen an welche wir uns halten werden

11.2 Leseauftrag

Lesen Sie im Tutorial die Abschnitte

- Datenbanken
- Access
- Namenkonventionen

Sie sollten Ihnen eine Übersicht über die Funktionen von Access bieten.

11.3 Notizen erfassen

Notieren Sie in OneNote die Antworten auf folgende Fragen:

- Welche Arten von Datenbankobjekten gibt es unter Access (6)?
- Beschreiben Sie die Datenbankobjekte kurz und knapp in eigenen Worten
- Beschreiben Sie was die *ungarische Namensnotation* ist
- Übernehmen Sie die Darstellung aus dem Tutorial (Liste aller Kürzel) in Ihre OneNote-Notizsammlung

11.4 Repetitionsvideo gucken

Das folgende Video geht nochmals durch die Access-Begriffe durch - kucken Sie es und passen Sie Ihre Notizen an.

Video-URL : <https://www.youtube.com/watch?v=L9W2gLpj728>

Access Kurs Teil 1 - Datenbank, Tabellen, Felder, Werte...



11.5 Selbsttest durchführen

Führen Sie den Selbsttest *Test - Namenskonvention und Access-Begriffe* auf Moodle aus. Sie dürfen Ihre Notizen verwenden! Aber kontrollieren Sie vielleicht dabei auch, ob Ihre Notizen “brauchbar” und nützlich sind.

Achtung: Der Selbsttest besteht aus vielen Fragen - Sie erhalten nur eine kleine Auswahl pro Durchgang zu sehen. Wollen Sie sicher gehen alle Fragen als Prüfungsvorbereitung gesehen zu haben, müssen Sie den Test mehrfach durchführen.

11.6 Lernzielkontrolle

Kontrollieren Sie bitte ob Sie die Lernziele erfüllen!

Jede/r Lernende:

- * ...weiss was Access ist und wofür es verwendet werden kann
- * ...kennt die Begriffe _Tabellen Abfragen Formulare Berichte Makros Module_ und kann sie :
- * ...kennt die Namenskonventionen an welche wir uns halten werden

12 MS-Access : Entitäten (Tabellen) und Attribute

Im folgenden Kapitel beschäftigen wir uns mit den Daten in Access selbst. Dazu müssen wir uns mit folgenden Dingen beschäftigen:

- Tabellen
- Attribute
- Datenimport und Datenexport

12.1 Lernziele

Jede/r Lernende:

- ...kennt die Attributtypen *Zahlen (Ganzzahlen, Fließkommazahlen, Exponentialzahlen)*, *Datum/Zeit inkl. Formate*, *Text (verschiedene Längen)*, *Boolean*, *AutoWert* und kann sie konfigurieren
- ...kann Beispieldaten in Access importieren
- ...kann aus Excel Daten in Access importieren
- ...kann Beispieldaten aus Access in Excel exportieren

12.2 Leseauftrag : Tabellen und Attribute

Lesen Sie im Tutorial den Abschnitt

- Eine Tabelle

Darin werden die Tabellen als “Datenhaltungs-Zentrum” vorgestellt. Auch werden die Attribute, im Tutorial Access 2007, vorgestellt.

12.3 Rechercheauftrag : Attribut-Typen

Lesen Sie unter

<https://support.office.com/de-de/article/einfuehrung-in-datentypen-und-feldeigenschaften-30ad644f-946c-442e-8bd2-be067361987c>

nach, welche Attribute unter Office 2016 zur Verfügung stehen.

12.4 Umsetzungsauftrag : Tabelle Erstellen

Erstellen Sie in Access in einer Testdatenbank eine neue Tabelle *tblAuto* mit folgenden Attributen:

- IDAutoID (AutoWert)
- lngKilometerLeistung (Long Int.)
- txtModellName (Text)
- txtHersteller (Text)
- txtVerkaufstext (Langer Text)
- dblBezinverbrauch (Double)
- datHerstellungsDatum (Format: Datum, lang)
- curVerkaufspreis (Währung)

Dokumentieren Sie Ihr Vorgehen für die eigenen Notizen.

12.5 Lernvideo als Repetition

Betrachten Sie als Vertiefung das folgende Video:

Video-URL : <https://www.youtube.com/watch?v=G4EdOoX6EOw>



12.6 Beispieldaten

12.6.1 Video : Beispieldaten importieren

Für unsere Arbeiten ist es oft wichtig, Beispieldaten zur Verfügung zu haben. In folgendem Video wird gezeigt, wie Sie Beispieldaten in Access importieren.

Video-URL : <https://www.youtube.com/watch?v=eRL1RE-tGFo>

Access 2016: Wie komme ich an Beispieldaten um Access zu lernen?



12.6.2 Umsetzungsauftrag

Vorgehen:

- Erstellen Sie in Access eine neue Datenbank *Testdaten*
- Erzeugen Sie über Migano 50 Testdatensätze mit den Standardwerten.
- Laden Sie die CSV-Datei herunter und speichern Sie die Datei
- Importieren Sie nun die Datei in Access nach Video-Anleitung...folgende Vorgaben
 - Übernehmen Sie das “Nr.-Attribut” als ID-Wert
 - Ändern Sie das “Nr.-Attribut” auf AutoWert

Dokumentieren Sie das Vorgehen für Ihre eigenen Notizen

12.7 Excel-Daten

12.7.1 Video : Excel-Daten importieren

Für die praktische Welt (in Ihrem Unternehmen), ist es wichtig Daten aus Excel in Access zu importieren. Das folgende Video zeigt, wie es geht:

Video-URL : <https://www.youtube.com/watch?v=8OCcBcCMv88>

Microsoft Access lernen: Quicktipp Excel-Daten importieren



12.7.2 Umsetzungsauftrag

Auftrag:

Sie haben auf Moodle eine Excel-Datei mit den All-Time Bestliste aller Filme. Importieren Sie diese Werte in eine Datenbank *Movies*.

Abschluss:

Dokumentieren Sie das Vorgehen für Ihre eignen Notizen

12.8 Exporte

12.8.1 Video : Daten aus Access in Excel exportieren

Manchmal müssen Sie auch Datensätze aus Access exportieren. Das folgende Lernvideo zeigt das Vorgehen:

Video-URL : <https://www.youtube.com/watch?v=nBWvwZMzYDY>



12.8.2 Umsetzungsauftrag

Auftrag:

Exportieren Sie Ihre *Testdaten*-Datenbank in eine Excel-Datei.

Abschluss:

Dokumentieren Sie das Vorgehen für Ihre eigenen Notizen



13 MS-Access : Mehrere Tabellen und Beziehungen

Sie haben bis hier viele nützliche Dinge zu MS-Access gelernt - lassen Sie uns das jetzt in diesem Kapitel erweitern. Alle bisherigen Schritte machen nur Sinn, wenn wir mehrere Tabellen miteinander in eine Relation (ER-Modell und ER-Diagramm) bringen können. Wir wollen uns also in diesem Kapitel mit folgenden Dingen auseinandersetzen:

- Primär- und Fremdschlüssel
- Relationen (1 x n)

13.1 Lernziele

Jede/r Lernende:

- ...kann die Begriffe *Index*, *Primärschlüssel*, *Fremdschlüssel* in eigenen Worten beschreiben
- ...kann einen *Index und Primärschlüssel* in Access konfigurieren
- ...kann 1:n - Beziehungen in Access zwischen zwei Tabellen konfigurieren
- ...kann den Begriff *referentielle Integrität* konfigurieren (über Beziehungen)
- ...kann eine *Löschweitergabe* auf Beziehungen konfigurieren

13.2 Leseauftrag : Schlüssel und Beziehungen

Lesen Sie im Tutorial den Abschnitt *Schlüssel*.

Machen Sie sich folgende Notizen:

- Beschreiben Sie in eigenen Worten was ein Index ist
- Beschreiben Sie in eigenen Worten was ein Primärschlüssel ist
- Beschreiben Sie in eigenen Worten was der Unterschied zwischen Index und Primärschlüssel ist
- Beschreiben Sie in eigenen Worten was ein Fremdschlüssel ist

13.3 Leseauftrag : Beziehungen

Lesen Sie im Tutorial den Abschnitt *Beziehungen*.

Machen Sie sich folgende Notizen:

- Beschreiben Sie in eigenen Worten, was die referentielle Integrität bedeutet
- Beschreiben Sie in eigenen Worten, wie ein Fremdschlüssel und ein Primärschlüssel in einer Beziehung genutzt werden
- Beschreiben Sie in eigenen Worten, was eine Löschweitergabe macht

13.4 Dokumentationsauftrag : Schlüssel, Index

Dokumentieren Sie in Ihren eigenen Unterlagen (mittels Bilder/Videos/Bemerkungen) wie:

- Ein Index gesetzt wird
- Ein Primärschlüssel gesetzt wird

13.5 Dokumentationsauftrag : Beziehungen

Dokumentieren Sie in Ihren eigenen Unterlagen (mittels Bilder/Videos/Bemerkungen) wie:

- Eine Beziehung zwischen zwei Tabellen erstellt wird (1:n)

- Welches jetzt der Fremdschlüssel und welcher der Primärschlüssel ist
- Wie Sie eine referentielle Integrität sicherstellen
- Wie Sie eine Löschweitergabe sicherstellen



13.6 Repetitionsvideo

Betrachten Sie als Repetition das folgende Video:

Video-URL : <https://www.youtube.com/watch?v=KIZLiGyRjDE>

Microsoft Access: Beziehungen in Datenbanken verstehen



14 Komplexe Beziehungen

Damit wir alle Beziehungsarten in unserer komplexen Umwelt abbilden können, reicht die Beziehungsart 1:n nicht aus - wir brauchen noch weitere und darum kümmern wir uns in diesem Kapitel.

Folgende Dinge wollen wir behandeln:

- n:m Beziehung
- 1:1 Beziehung
- Selbstbezug
- Spezialfälle : c:c, mc:mc, mc:c

14.1 Lernziele

Jede/r Lernende:

- ...kann das Vorgehen bei der Umsetzung einer n:m Beziehung in der Praxis in eigenen Worten beschreiben
- ...kann eine vorgegebene n:m Beziehung in Access umsetzen
- ...kann die Anwendung eines *zusammengesetzter Schlüssel* bei n:m Beziehung erklären
- ...kann eine vorgegebene 1:1 Beziehung in Access umsetzen
- ...kann in eigenen Worten beschreiben was ein Selbstbezug ist
- ...kann einen Selbstbezug in Access umsetzen

14.2 n:m Beziehung

14.2.1 Leseauftrag

Lesen Sie im Tutorial den Abschnitt *n:m Beziehung*.

Machen Sie sich folgende Notizen:

- Beschreiben Sie in eigenen Worten, wann und wozu eine Verknüpfungstabelle erstellt wird

14.2.2 Theorie-Video

Video-URL : <https://youtu.be/WtI-hQ2Dp1I>

am5123 m:n-Beziehungen zwischen Datenbank-Tabellen - Teil 1



14.2.3 Praxis-Video (bis ca. 15min / ohne Formulare)

Video-URL : <https://youtu.be/6uejqv9tD2E>



14.2.4 Dokumentationsauftrag

Dokumentieren Sie in Ihren eigenen Unterlagen (mittels Bilder/Videos/Bemerkungen) wie:

- Sie im Schulbeispiel (Musikbibliothek) eine n:m Beziehung umsetzen

14.3 1:1 Beziehung

14.3.1 Leseauftrag

Lesen Sie im Tutorial den Abschnitt *1:1 Beziehung*.

Die Erklärung im Tutorial ist nicht in allen Stellen glücklich...versuchen wir es mit folgender Zusammenfassung:

Eine 1:1 Beziehung ist nichts anderes als ein Spezialfall der 1:n Beziehung, bei der es verboten ist, dass andere Datensätze den selben Datensatz aus der anderen Tabelle referenzieren <- Ergo: Müssen Duplikate verboten werden!



Machen Sie sich folgende Notizen:

- Beschreiben Sie in eigenen Worten, was eine 1:1 Beziehung genau ist

14.4 Selbstbezug

14.4.1 Leseauftrag

Lesen Sie im Tutorial den Abschnitt *Selbstbezug*.

Machen Sie sich folgende Notizen:

- Übernehmen Sie das Beispiel aus dem Tutorial in Ihre Notizen und ergänzen Sie es mit eigenen Bemerkungen

14.5 Spezialfälle (c:c, mc:mc, mc:c)

Wir haben in der Zeit vor Access immer noch die Spezialfälle mit “c” behandelt. Dabei ging es eigentlich immer darum, ob ein Datensatz auch 0 sein kann. All diese Spezialfälle lösen Sie eigentlich damit, in dem Sie einem Fremdschlüssel-Attribut erlauben 0 zu sein.

In Access heisst diese Option auf dem Feld “Eingabe erforderlich”.

Zusammenfassend...wollen Sie:

Eine c:c-Beziehung bauen:

...müssen Sie eine Fremdschlüssel-Primärschlüssel-Beziehung bauen. Das Fremdschlüssel-Attribut darf leer sein, erlaubt aber keine Duplikate.

Eine mc:mc-Beziehung bauen:

...müssen Sie eine n:m-Beziehung bauen, mit einer Verknüpfungstabelle. Die Fremdschlüssel-Attribute auf den zu verknüpfenden Tabellen dürfen aber leer sein.

Eine mc:c-Beziehung bauen:

...müssen Sie eine Fremdschlüssel-Primärschlüssel-Beziehung bauen. Das Fremdschlüssel-Attribut darf leer sein und Duplikate sind erlaubt



15 Normalisierung / Theorie

Es ist an sich relativ selten, dass man auf "einer grünen Wiese" eine Datenbanklösung bauen kann. Typischerweise erhält man eine Tabelle mit ungeordneten Daten, die man dann in eine Datenbank überführen soll.

In diesem Kapitel erlernen Sie, wie Sie auf jeden Fall zu einem brauchbaren Datenbankdesign kommen werden. Dazu benötigt es allerdings ein bisschen Theorie: Die Normalisierung.

15.1 Lernziele

Jede/r Lernende:

- ...kann die 0. bis 3. Normalform anhand von Beispielen erklären
- ...kann Datensätze in die 0. bis 3. Normalform überführen

15.2 0. und 1. Normalform

Leseauftrag:

Lesen Sie im Tutorial im Abschnitt *Normalformen* die ersten beiden Abschnitte (0. und 1. Normalform).

Beantworten Sie in Ihren Notizen folgende Fragen:

- Was ist bei einer Datenbank falsch, wenn sie nicht der 0. Normalform entspricht?
- Was muss man tun um die 0. Normalform zu erreichen?
- Was ist bei einer Datenbank falsch, wenn sie nicht der 1. Normalform entspricht?
- Was muss man tun um die 1. Normalform zu erreichen?

Betrachten Sie folgendes Video:

- <https://www.youtube.com/watch?v=sFG5pR5016k>

Datenbanken - Normalisierung - 1. Normalform



15.3 2. Normalform

Leseauftrag:

Lesen Sie im Tutorial im Abschnitt *Normalformen* den dritten Abschnitt (2. Normalform).

Beantworten Sie in Ihren Notizen folgende Fragen:

- Was ist bei einer Datenbank falsch, wenn sie nicht der 2. Normalform entspricht?
- Was muss man tun um die 2. Normalform zu erreichen?

Betrachten Sie folgendes Video:

- <https://www.youtube.com/watch?v=kAuZ-v-HBtA>

Datenbanken - Normalisierung - 2. Normalform



15.4 3. Normalform

Leseauftrag:

Lesen Sie im Tutorial im Abschnitt *Normalformen* den vierten Abschnitt (3. Normalform).

Beantworten Sie in Ihren Notizen folgende Fragen:

- Was ist bei einer Datenbank falsch, wenn sie nicht der 3. Normalform entspricht?
- Was muss man tun um die 3. Normalform zu erreichen?

Betrachten Sie folgendes Video:

- <https://www.youtube.com/watch?v=x183UkdsPQ8>

Datenbanken - Normalisierung - 3. Normalform



16 Normalisierung / Praxis¹

Betrachten Sie folgendes Beispiel einer Liste:

Name	Anschrift	Wohnort	Rechnungsnummer und -datum		Artikel	Rechnungsbetrag
Gustav Meier-Senn	Im Tobel 13	9248 Bichwil	Nr 001	01.02.2004	5, 8, 8, 8, 4, 9	SFr. 358.00
Ida Kuhler	Walchenstr. 45b	5224 Unterbözberg	Nr 002	01.02.2004	7,7,7	SFr. 90.00
Gustav Meier-Senn	Im Tobel 13	9248 Bichwil	Nr 004	20.03.2004	7,5,2,5,9	SFr. 750.00
Sylvia Brendel	Rennweg 7	8000 Zürich				
Gustav Meier-Senn	Im Tobel 13	9248 Bichwil	Nr 007	21.03.2004	8, 4, 5	SFr. 320.00
Morani	Birkenweg 16	1265 La Cure	Nr 008	22.03.2004	4, 8	SFr. 40.00
Sylvia Benz	Im Tobel 13	5600 Baden				
Walter Morani	Birkenweg 16	1265 La Cure	Nr 009	22.03.2004	9,1	SFr. 42.00
Merz, Beat	Seehalde 34	5600 Lenzburg	Nr 010	25.03.2004	6, 6, 9,8	SFr. 520.00

Was kann schief gehen, wenn...

- sie die Adresse von Gustav Meier-Senn auf «Im Tobel 18» korrigieren
- Walter Morani aus La Cure per heutigem Datum eine neue Rechnung bekommt, weil er die Artikel 4, 6, 2 und 9 gekauft hat
- Ida Kuhler alle Artikel der Rechnung Nr. 002 zurückgegeben hat
- Walter Morani den Artikel Nummer 9 von der Rechnung Nummer 009 zurückgibt

Notieren sie mögliche Fehlerquellen im Datenbestand, die bei diesen Änderungen entstehen können.

Ganz offenbar liegen diese Daten so nicht in einem optimalen Zustand vor. Es ist unser Job das zu ändern.

16.1 Auftrag 1 - Negativbeispiel

Teilen Sie die Tabelle aus dem vorhergehenden Abschnitt so auf, dass keine redundanten Daten mehr abgespeichert sind. Zeichnen sie für ihre Lösung das entsprechende ERD auf und erstellen sie eine Tabellenbeschreibung.

16.2 Auftrag 2 - Autohandel

Ein Autohändler hat seine Verkäufe bisher in einer Excel-Tabelle erfasst. Führen sie in dieser Tabelle eine Normalisierung durch und erstellen sie das ERD und die Tabellenbeschreibung.

- [Excelsheet mit den Daten](#)

Kundenname	Adresse	Automarke	Typ	Seriennummer	Verkäufer	Datum
Meier	Planetenweg 7	VW	Golf	123456	Schmid	23.4.12
		Opel	Kadett	345678	Plüss	7.8.12
Müller	Altstadt 12	VW	Golf	388721	Frey	17.6.12
Steffen	Gartenstr. 7	VW	Polo	222245	Schmid	15.7.12
Steffen	Augasse 12	Audi	Quattro	122154	Frey	13.11.12
		Opel	Manta	445321		
					Schenk	

16.3 Auftrag 3 - Reisebüro

Führen sie in dieser Tabelle eine Normalisierung durch und erstellen sie das ERD und die Tabellenbeschreibung. Beachten sie, dass es möglich ist, Hinflüge ohne Rückflüge zu buchen. Ein Hin- bzw. Rückflug kann sich aus mehreren Teilflügen zusammensetzen.

- [Excelsheet mit den Daten](#)

Buchungsdatum	Preis CHF	Name	Vorname	Adresse	Ort	Reiseziel	Hotel	Anzahl Personen	Hinflug	Hinflugdatum	Hinflugzeit	Rückflug	Rückflugdatum	Rückflugzeit
12.12.12	2450.-	Müller	Hugo	Saturnweg 7	Laufen	Rio	Hilton	2	SR220	12.3.13	07:15	BA321	15.3.13	12:10
22.12.12	450.-	Meier	Max	Feldweg 5	Buckten	Birmingham	Royal	1	BA212	23.4.13	8:20	SR212	28.4.13	12:30
									SR420	23.4.13	9:20			
1.1.13	4450.-	Schmid	Beate	Hauptstr. 13	Aesch	Hawaii	Aloha	3	SR212	25.5.13	12:40	-	-	-
4.1.13	840.-	Müller	Hugo	Saturnweg 7	Laufen	Frankfurt	Tropica	4	BA123	12.3.13	12:10	DA110	12.4.13	21:10
15.1.13	1820.-	Steffen	Felix	Heuboden 2	Pratteln	St. Domingo	Royal	1	AF320	24.5.13	8:15	AF210	4.6.13	9:30
									AF512	24.5.13	17:20	AF212	4.6.13	18:20
1.2.13	2400.-	Müller	Hugo	Flühstr. 12	Reinach	Caracas	Cetral	2	AV555	12.4.13	10:00	CR101	5.6.13	7:20
									VH113	12.4.13	22:30	-	-	-
						Rio	Pallas							
						Ibiza	Perle Mango							

16.4 Auftrag 4 - Reisebüro

Eine Schulbibliothek hat den Wunsch, alle Bücher über eine Datenbank zu verwalten. Dabei interessiert folgendes:

Vom Kunden, das ist die Person, die Bücher ausleiht, seinen Name, Vorname, Adresse, PLZ, und Wohnort Bei den Büchern der/die Buchautor(en), Buchtitel, Genre (Themen), ISBN-Nummer, Jahrgang und den Bücherstandort Von den ausgeliehenen Bücher das Ausleihdatum, Rückgabedatum und die Leihgebühr Jedes Buch wird einem «Genre» (Buchgattung, Thema)

zugeordnet, wobei auch Genres erfasst werden können, von welchen noch keine entsprechenden Bücher vorhanden sind. Ein Autor kann mehrere Bücher schreiben, ein Buch aber auch mehrere Autoren haben. Jeder Kunde wird beim Ausleihen und der Rückgabe im System mutiert. Es können vom gleichen Titel mehrere Bücher vorhanden sein.

Auftrag: Entwerfen sie eine geeignete Datenbasis. Zeichnen sie das entsprechende ERD und erstellen sie die Tabellenbeschreibungen. Es dürfen übrigens zu keinem Zeitpunkt Mehrfacheinträge oder Nullwerte vorkommen.

1. Die Inhalte in diesem Kapitel wurden von

http://edu.juergarnold.ch/modul_104/normalisieren/article.html übernommen. Auf jener Seite finden Sie auch die Musterlösungen↩

17 Formulare in Access

17.1 Lernziele

Jede/r Lernende:

- ...kann in Access eigene Formulare erstellen

17.2 Repetition : Daten importieren

Betrachten Sie als Repetition das Video von unten. Es geht darum, Beispieldaten zu importieren - führen Sie alle Schritte durch.

Video-URL : <https://www.youtube.com/watch?v=eRL1RE-tGFo>

Access 2016: Wie komme ich an Beispieldaten um Access zu lernen?



17.3 Video- und Mitmach-Auftrag

Video-URL : <https://www.youtube.com/watch?v=L-phP3R7UFg>

Access: So erstellst du Formulare für Tabellen oder Menüs



17.4 Leseauftrag

Lesen Sie das Kapitel “Steuerelemente” aus dem Access-Tutorial. Dokumentieren Sie für Ihre eigenen Unterlagen welche Steuerelemente in Access zur Verfügung stehen.

17.5 Dokumentationsauftrag

Dokumentieren Sie für Ihre eigenen Unterlagen wie Sie ein Formular erstellen und anpassen können.

18 Exkurs : HTML, PHP

Wir werden im folgenden eine datenbank-gestützte Webanwendung in Betrieb nehmen. Die Inhalte aus diesem Exkurs (dieses und die folgenden Kapitel) gehen weit über die geforderten Inhalte aus dem Modul 104 hinaus. Ziel soll es sein, Ihnen eine konkrete Anwendung der Inhalte aus dem Modul zu zeigen. Gleichzeitig sollen Sie auch eine Einführung in SQL erhalten.

Wir arbeiten in diesem Exkurs mit einer Linux-VM, auf welcher die folgenden Dinge bereits vorinstalliert sind:

- Apache2-Webserver
- Programmiersprache php
- Die Datenbanksoftware MySQL
- Die Software *PhpMyAdmin*

18.1 Zugriffe

Die Passwörter für die VM lauten:

VM:

bztadmin / admin -> sudo-Passwort: admin

Datenbank-Root:

root / PASSWORD

18.2 Wie arbeiten

Linux ist ein echtes Mehrbenutzer-Betriebssystem, was zur Folge hat das Datei- und Ordnerberechtigungen ganz genau umgesetzt werden. Wir werden mit dem Texteditor *Atom* Dateien und Ordner für den Webserver erstellen und bearbeiten. Damit das funktioniert, müssen Sie den Editor Atom als *root laufen lassen*:

```
sudo atom
```

18.3 HTML und CSS und PHP?

Ein paar Begriffserklärungen vorne weg - was ist (nach Wikipedia):

HTML:

HTML ist eine Abkürzung für Hypertext Markup Language. Auf Deutsch bedeutet dies so viel wie „Auszeichnungssprache für verknüpften Text“. HTML ist heute der wichtigste Auszeichnungsstandard im Internet. Er ist Voraussetzung für die Programmierung und das Design von Webinhalten. Andere Standards wie PHP bauen in erheblichem Maße auf HTML auf.

CSS:

Cascading Style Sheets, kurz CSS genannt, ist eine Stylesheet-Sprache für elektronische Dokumente und zusammen mit HTML und DOM eine der Kernsprachen des World Wide Webs. Sie ist ein sogenannter „living standard“ (lebendiger Standard) und wird vom World Wide Web Consortium (W3C) beständig weiterentwickelt. Mit CSS werden Gestaltungsanweisungen erstellt, die vor allem zusammen mit den Auszeichnungssprachen HTML und XML (zum Beispiel bei SVG) eingesetzt werden.

PHP:

PHP (rekursives Akronym und Backronym für „PHP: Hypertext Preprocessor“, ursprünglich „Personal Home Page Tools“) ist eine Skriptsprache mit einer an C und Perl angelehnten Syntax, die hauptsächlich zur Erstellung dynamischer Webseiten oder Webanwendungen verwendet wird. PHP wird als freie Software unter der PHP-Lizenz verbreitet. PHP zeichnet sich durch breite Datenbankunterstützung und Internet-Protokolleinbindung sowie die Verfügbarkeit zahlreicher Funktionsbibliotheken aus.

18.4 Action : GUI bauen

Wir gehen im folgenden, ohne grosse Erklärungen, durch die GUI-Erstellung durch. Öffnen Sie dazu den Atom-Editor (wie beschrieben) und arbeiten Sie im Verzeichnis `/var/www/html`.

18.4.1 Schritt 1 : Startseite erstellen

Wir benötigen eine Startseite - erstellen Sie eine Datei `/var/www/html/index.php` mit folgendem Inhalt:

```
<!doctype html>
<html lang="en">

<head>
  <meta charset="utf-8">
  <meta http-equiv="x-ua-compatible" content="ie=edge">
  <meta name="viewport" content="width=device-width, initial-scale=1">

  <title>Simple Database App</title>

  <link rel="stylesheet" href="css/style.css">
</head>

<body>
  <h1>Simple Database App</h1>

  <ul>
    <li><a href="create.php"><strong>Create</strong></a> - add a user</li>
    <li><a href="read.php"><strong>Read</strong></a> - find a user</li>
  </ul>

</body>
</html>
```

Testen:

Öffnen Sie im Browser die URL `_http://localhost/index.php_` - Sie sollten Ihre Seite sehen

18.4.2 Schritt 2 : Benutzer erstellen können

Wir benötigen eine neue Seite `create.php`, damit wir für unsere Applikation Benutzer erfassen können - erstellen Sie eine Datei `/var/www/html/create.php` mit folgendem Inhalt:

```
<!doctype html>
<html lang="en">

<head>
    <meta charset="utf-8">
    <meta http-equiv="x-ua-compatible" content="ie=edge">
    <meta name="viewport" content="width=device-width, initial-scale=1">

    <title>Simple Database App</title>

    <link rel="stylesheet" href="css/style.css">
</head>

<body>
    <h1>Simple Database App</h1>

    <form method="post">
        <label for="firstname">First Name</label>
        <input type="text" name="firstname" id="firstname">
        <label for="lastname">Last Name</label>
        <input type="text" name="lastname" id="lastname">
        <label for="email">Email Address</label>
        <input type="text" name="email" id="email">
        <label for="age">Age</label>
        <input type="text" name="age" id="age">
        <label for="location">Location</label>
        <input type="text" name="location" id="location">
        <input type="submit" name="submit" value="Submit">
    </form>

    <a href="index.php">Back to home</a>

</body>
</html>
```

Testen:

Öffnen Sie im Browser die URL `_http://localhost/create.php_` - Sie sollten Ihre Seite sehen. Ebenfalls sollte der Link unten funktionieren.

18.5 Schritt 3 : Wir benötigen Style...

Wir sind uns einig: Die Webseite sieht noch sehr unschön aus. Versuchen wir das zu verbessern.

Wir benötigen eine Style-Sheet-Datei `style.css` - erstellen Sie eine Datei

`/var/www/html/css/style.css` mit folgendem Inhalt:

```
label {  
    display: block;  
    margin: 5px 0;  
}
```

Testen:

Rufen Sie die Ihre Webseiten `index.php` und `create.php` nochmals auf...sie sollten jetzt ein bisschen besser aussehen.

18.5.1 Schritt 4 : Benutzer auslesen können

Wir benötigen eine neue Seite `read.php`, damit wir für unsere Applikation Benutzer lesen/darstellen können - erstellen Sie eine Datei `/var/www/html/read.php` mit folgendem Inhalt:

```
<!doctype html>
<html lang="en">

<head>
  <meta charset="utf-8">
  <meta http-equiv="x-ua-compatible" content="ie=edge">
  <meta name="viewport" content="width=device-width, initial-scale=1">

  <title>Simple Database App</title>

  <link rel="stylesheet" href="css/style.css">
</head>

<body>
  <h1>Simple Database App</h1>

  <h2>Find user based on location</h2>

  <form method="post">
    <label for="location">Location</label>
    <input type="text" id="location" name="location">
    <input type="submit" name="submit" value="View Results">
  </form>

  <a href="index.php">Back to home</a>

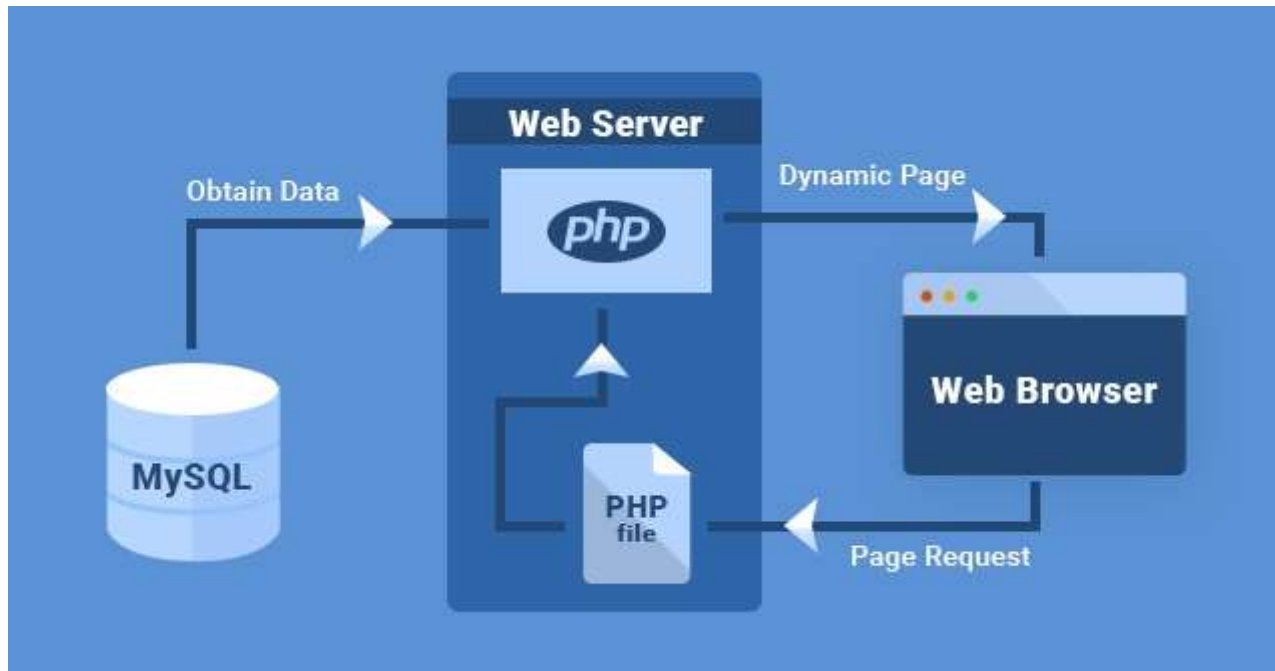
</body>
</html>
```

18.6 Abschluss erste Schritte

Sie sollten bis hier hin eine funktionierende Webseite haben, bei der alle Links funktionsfähig sind. Die nächsten Schritte beinhalten dann die Bereitstellung der Datenbank.

18.7 Erklärung zu den Server-Diensten

Damit eine datenbankgestützte Webapplikation funktioniert, müssen viele Dinge gemeinsam funktionieren. Die folgende Abbildung stellt das dar:



Bemerkung:

In unserem Beispiel ist es einfach so, dass die Datenbank auf der selben VM läuft, wie der Webserver. Das ist in der Praxis eher selten der Fall.

Wichtig ist:

PHP ist eine Programmiersprache, die vom Webserver ausgeführt wird. Die Aufgabe von PHP ist es, eine HTML-Datei so zu erstellen, dass sie sinnvolle Informationen enthält. Die HTML-Datei wiederum wird dann in Ihrem Webbrowser dargestellt.

19 Exkurs : SQL und phpmyadmin

Im folgenden Kapitel wenden wir uns der Datenbank zu. Dabei lernen wir eine neue Sprache (keine Programmiersprache, sondern eine *Abfrage-Sprache*) kennen: SQL. SQL ist im Umgang mit Datenbank zentral. Zuerst mal eine Definition nach Wikipedia:

SQL:

SQL ist eine Datenbanksprache zur Definition von Datenstrukturen in relationalen Datenbanken sowie zum Bearbeiten (Einfügen, Verändern, Löschen) und Abfragen von darauf basierenden Datenbeständen. Die Sprache basiert auf der relationalen Algebra, ihre Syntax ist relativ einfach aufgebaut und semantisch an die englische Umgangssprache angelehnt. Ein gemeinsames Gremium von ISO und IEC standardisiert die Sprache unter Mitwirkung nationaler Normungsgremien wie ANSI oder DIN. Fast alle gängigen Datenbanksysteme unterstützen SQL – allerdings in unterschiedlichem Umfang und leicht voneinander abweichenden „Dialekten“. Durch den Einsatz von SQL strebt man die Unabhängigkeit der Anwendungen vom eingesetzten Datenbankmanagementsystem an.

Zusammenfassend kann man sagen: Mit SQL ersetzen Sie das Geklicke unter MS Access um Tabellen, Datenbanken etc. zu erstellen, abzufragen und zu bearbeiten.

19.1 Werkzeug: PhpMyAdmin

Obwohl wir auch über die Konsole mit der Datenbank arbeiten könnten (was in der Praxis oft/ausschliesslich der Fall ist), werden wir mit einem Werkzeug arbeiten das uns die Arbeit etwas vereinfacht: PhpMyAdmin.

Testen:

Testen Sie bitte den Zugriff auf die Webseite http://localhost/phpmyadmin_ - Sie sollten eine Webseite mit einem Login erhalten. Melden Sie sich als Datenbank-Root an.

19.2 Schritt 1 : Datenbank erstellen

Ein relationales Datenbank-Management-System arbeitet also mit SQL um Dinge zu erstellen/löschen/bearbeiten. In einem ersten Schritt wollen wir eine neue Datenbank erstellen. Wählen Sie dazu oben in der GUI den Tab “SQL” aus.

Kopieren Sie folgenden Code in das Freitextfeld und führen Sie den Code aus:

```
CREATE DATABASE usertest;

use usertest;
```

Testen:

Sie sollten jetzt links in der Darstellung eine neue Datenbank erkennen. Klicken Sie nun links auf Ihre Datenbank.

19.3 Schritt 2 : Tabelle erstellen

Um nun Daten in einer Datenbank zu speichern, benötigen wir Tabellen mit Attributen. Achtung, Sie müssen:

- ...links im Baum auf Ihre Datenbank klicken
- ...nochmals oben im Tab “SQL” anklicken

Kopieren Sie folgenden Code in das Freitextfeld und führen Sie den Code aus:

```
CREATE TABLE users (
    id INT(11) UNSIGNED AUTO_INCREMENT PRIMARY KEY,
    firstname VARCHAR(30) NOT NULL,
    lastname VARCHAR(30) NOT NULL,
    email VARCHAR(50) NOT NULL,
    age INT(3),
    location VARCHAR(50),
    date TIMESTAMP
);
```

Testen:

Sie sollten jetzt links im Baum eine neue Tabelle sehen.

19.4 Schritt 3 : Daten hinzufügen

Im Moment ist Ihre Tabelle noch leer...Sie dürfen das gerne austesten, in dem Sie im SQL-Feld folgende Abfrage ausführen:

```
SELECT * from users;
```

Lassen Sie uns Daten erfassen. Kopieren Sie den folgenden Code in das SQL-Feld und führen Sie den Code aus:

```
INSERT INTO `users` (`id`, `firstname`, `lastname`, `email`, `age`, `location`, `date`)  
VALUES (NULL, 'Christian', 'Inauen', 'christian.inauen@bztbf.ch', '27', 'Tobel', CURRENT_TIMESTAMP);
```

Testen:

Führen Sie anschliessend nochmals den Befehl `SELECT * from users;` aus - sollte anders aussehen als vorher - nicht?

19.4.1 Auftrag: Benutzer erfassen

Auftrag:

Füllen Sie nun Ihre Datenbank anhand des Beispiels von oben (min. 5 Benutzer) - Passen Sie einzelne Werte an, andere belassen Sie gleich.

19.5 Schritt 4 : Daten auslesen

Der Befehl

```
SELECT * from users;
```

ist die "Dampfhammer-Methode" um Daten auszulesen....es wird einfach alles (*) geliefert. Häufig möchte man aber gefilterte Datensätze auslesen...zum Beispiel alle Benutzer mit dem Vornamen "Christian". Das macht man so:


```
SELECT * FROM `users` WHERE firstname="Christian";
```

Man kann auch mehrere Filter setzen:

```
SELECT * FROM `users` WHERE firstname="Christian" AND lastname="Inauen";
```

Dabei werden von der Datenbank aber immer auch alle Spalten/Attribute ausgelesen, auch solche die man nicht anzeigen will. Auch hier kann man bei SQL sagen was man gerne angezeigt haben möchte. Dazu müssen Sie einfach den * ersetzen, mit den Spalten die Sie angezeigt haben möchten. Beispiel: Wir wollen nur die Email-Adresse von allen Benutzern mit dem Vornamen Christian anzeigen - dann geht das so:

```
SELECT email FROM `users` WHERE firstname="Christian";
```

20 Exkurs : Erweiterung PHP und Formulare

So - nachdem wir jetzt die Datenbank und GUI vorbereitet haben, kommt jetzt natürlich der heikle Teil: Wir wollen die GUI so programmieren, damit wir die Nutzererfassung über die Web-GUI erledigen können. Aber wie vorher: Schritt für Schritt.

Im Wesentlichen funktioniert es jetzt so, dass wir mit den Daten, die wir aus dem Formular auslesen, einen schönen SQL-Befehl zusammenbauen und diesen Befehl dann auf die Datenbank ausführen - und das Ganze mittels PHP umgesetzt.

20.1 Schritt 1 : Datenbankverbindung mit PHP herstellen

Damit PHP überhaupt etwas von der Datenbank weiss, muss zuerst eine Verbindung hergestellt werden. Kopieren Sie bitte folgendes Code-Fragment:

```
<?php
$servername = "localhost";
$username = "root";
$password = "password";
$dbase = "usertest";

// Create connection
$conn = new mysqli($servername, $username, $password, $dbase);

// Check connection
if ($conn->connect_error) {
    die("Connection failed: " . $conn->connect_error);
}
?>
```

und kopieren Sie es bei create.php und read.php oben ein. Die Dateien sollten dann wie folgt aussehen:

create.php



```
<!doctype html>
<html lang="en">

<head>
    <meta charset="utf-8">
    <meta http-equiv="x-ua-compatible" content="ie=edge">
    <meta name="viewport" content="width=device-width, initial-scale=1">

    <title>Simple Database App</title>

    <link rel="stylesheet" href="css/style.css">
</head>

<body>
    <h1>Simple Database App</h1>

    <?php
    $servername = "localhost";
    $username = "root";
    $password = "password";
    $database = "usertest";

    // Create connection
    $conn = new mysqli($servername, $username, $password, $database);

    // Check connection
    if ($conn->connect_error) {
        die("Connection failed: " . $conn->connect_error);
    }
    ?>

    <form method="post">
        <label for="firstname">First Name</label>
        <input type="text" name="firstname" id="firstname">
```

```
<label for="lastname">Last Name</label>
<input type="text" name="lastname" id="lastname">
<label for="email">Email Address</label>
<input type="text" name="email" id="email">
<label for="age">Age</label>
<input type="text" name="age" id="age">
<label for="location">Location</label>
<input type="text" name="location" id="location">
<input type="submit" name="submit" value="Submit">
</form>

<a href="index.php">Back to home</a>

</body>
</html>
```

read.php

```
<!doctype html>
<html lang="en">

<head>
    <meta charset="utf-8">
    <meta http-equiv="x-ua-compatible" content="ie=edge">
    <meta name="viewport" content="width=device-width, initial-scale=1">

    <title>Simple Database App</title>

    <link rel="stylesheet" href="css/style.css">
</head>

<body>
    <h1>Simple Database App</h1>

    <h2>Find user based on location</h2>

    <?php
    $servername = "localhost";
    $username = "root";
    $password = "password";
    $database = "usertest";

    // Create connection
    $conn = new mysqli($servername, $username, $password, $database);

    // Check connection
    if ($conn->connect_error) {
        die("Connection failed: " . $conn->connect_error);
    }
    ?>

    <form method="post">
        <label for="location">Location</label>
```

```
<input type="text" id="location" name="location">
<input type="submit" name="submit" value="View Results">
</form>

<a href="index.php">Back to home</a>

</body>
</html>
```

20.2 Schritt 2 : Formular auslesen

Jetzt ist es so, dass wenn bei den Formularen der Button gedrückt wird ("Submit" oder ähnliches), die Formulardaten ausgelesen werden und an den Webserver übermittelt werden. Innerhalb vom PHP kann man dann auf diese Werte zugreifen....Beispiel:

```
<?php
    if(isset($_POST["submit"])){

        echo "Die Variablen firstname: " . $_POST["firstname"] . " und lastname " . $_POST["lastname"];

        $conn->close();
    }
```

Erklärung:

Wenn jetzt also "submit" gedrückt wird, wird die Variable `$_POST["submit"]` gesetzt. Die ganze Webseite wird nochmals ausgeführt und nun können die Variablen

`$_POST["firstname"]` und `$_POST["lastname"]` ausgelesen werden

20.3 Schritt 3 : ...SQL bauen mit Variablen und Ausführen (Insert)

Nachdem wir jetzt wissen wie solche Variablen genutzt werden, müssen wir mit den Werten einen SQL-Befehl zusammenbauen, der so funktioniert wie im vorhergehenden Kapitel. Betrachten Sie folgenden Code:

```
<?php
    if(isset($_POST["submit"])){

        $firstname = $_POST['firstname'];
        $lastname = $_POST['lastname'];
        $email = $_POST['email'];
        $age = $_POST['age'];
        $location = $_POST['location'];

        $sql = "INSERT INTO users (firstname, lastname, email, age, location) VALUES (

        if ($conn->query($sql) === TRUE) {
            echo "<script type= 'text/javascript'>alert('New record created successful'";
        } else {
            echo "<script type= 'text/javascript'>alert('Error: " . $sql . "<br>" . $conn->error());
        }

        $conn->close();
    }
?>
```

Fügen Sie den Code von oben so in Ihr create.php-Script ein, dass Ihr Script wie folgt aussieht:


```
<!doctype html>
<html lang="en">

<head>
    <meta charset="utf-8">
    <meta http-equiv="x-ua-compatible" content="ie=edge">
    <meta name="viewport" content="width=device-width, initial-scale=1">

    <title>Simple Database App</title>

    <link rel="stylesheet" href="css/style.css">
</head>

<body>
    <h1>Simple Database App</h1>

    <?php
    $servername = "localhost";
    $username = "root";
    $password = "password";
    $database = "usertest";

    // Create connection
    $conn = new mysqli($servername, $username, $password, $database);

    // Check connection
    if ($conn->connect_error) {
        die("Connection failed: " . $conn->connect_error);
    }
    ?>

    <form method="post">
        <label for="firstname">First Name</label>
        <input type="text" name="firstname" id="firstname">
```

```
<label for="lastname">Last Name</label>
<input type="text" name="lastname" id="lastname">
<label for="email">Email Address</label>
<input type="text" name="email" id="email">
<label for="age">Age</label>
<input type="text" name="age" id="age">
<label for="location">Location</label>
<input type="text" name="location" id="location">
<input type="submit" name="submit" value="Submit">
</form>
```

```
<?php
```

```
    if(isset($_POST["submit"])){
```

```
        $firstname = $_POST['firstname'];
```

```
        $lastname = $_POST['lastname'];
```

```
        $email = $_POST['email'];
```

```
        $age = $_POST['age'];
```

```
        $location = $_POST['location'];
```

```
        $sql = "INSERT INTO users (firstname, lastname, email, age, location) VALUES (
```

```
        if ($conn->query($sql) === TRUE) {
```

```
            echo "<script type= 'text/javascript'>alert('New record created successful
```

```
        } else {
```

```
            echo "<script type= 'text/javascript'>alert('Error: " . $sql . "<br>" . $co
```

```
        }
```

```
        $conn->close();
```

```
    }
```

```
?>
```

```
<a href="index.php">Back to home</a>
```

```
</body>
</html>
```



Testen:

Sie sollten jetzt über die GUI Benutzer erfassen können - kontrollieren Sie auch ob die Daten in der Datenbank korrekt ankommen.

20.4 Schritt 4 : ...SQL bauen mit Variablen und Ausführen (Read)

Abschliessend zu diesem Kapitel wollen wir natürlich noch die Benutzer auf der Webseite anzeigen können. SQL-technisch keine Herausforderung, dafür wird die GUI-Darstellung etwas spezieller...

Zielformulierung:

Wir wollen Benutzer, die an einer bestimmten Location wohnen, in einer HTML-Tabelle anzeigen können

20.4.1 CSS erweitern

Um Tabellen (einigermassen) schön darzustellen, erweitern Sie bitte das

`/var/www/html/style.css` um folgende neuen Informationen:

```
table {
    border-collapse: collapse;
    border-spacing: 0;
}

td,
th {
    padding: 5px;
    border-bottom: 1px solid #aaa;
}
```

20.4.2 Tabellen in HTML

Ich weiss nicht ob Sie es wissen, aber Tabellen werden in HTML wie folgt dargestellt:

```
<table>
<!-- Hier beginnt der Kopf/Ueberschrift der Tabelle -->
  <thead>
    <tr>
      <th>First Name</th>
      <th>Last Name</th>
      <th>Email Address</th>
    </tr>
  </thead>

  <!-- Hier kommen die Nutzdaten Tabelle -->
  <tbody>
    <!--Zeile 1-->
    <tr>
      <td>Christian</td>
      <td>Test</td>
      <td>test@test</td>
    </tr>
    <!--Zeile 2-->
    <tr>
      <td>Roger</td>
      <td>Test</td>
      <td>roger@test</td>
    </tr>
  </tbody>
</table>
```

Die Idee ist es jetzt natürlich, dass wir mittels PHP die Nutzdaten (Benutzer) auslesen und jeden Benutzer eine eigene neue Zeile erstellen (<- hört sich verdächtig nach Schleifen an...).

20.4.3 Alles zusammen

Folgender Plan:

1. Filter-Wert aus Formular auslesen und in Variable speichern
2. Mit Filter-Wert einen SQL-Select bauen
3. SQL-Select ausführen....dann erhalten wir einen Array von gültigen Werten als Resultat
4. Mittel Schleife durch den Array gehen und die Tabelle zusammenbauen

Los:



```
<?php
```

```
if(isset($_POST["submit"])){
```

```
    $location = $_POST['location'];
```

```
    $sql = "SELECT * from users WHERE location='".$location."'";
```

```
    //echo $sql;
```

```
    $result = $conn->query($sql);
```

```
    if ($result->num_rows > 0) {?
```

```
        <h2>Results</h2>
```

```
        <table>
```

```
            <thead>
```

```
                <tr>
```

```
                    <th>#</th>
```

```
                    <th>First Name</th>
```

```
                    <th>Last Name</th>
```

```
                    <th>Email Address</th>
```

```
                    <th>Age</th>
```

```
                    <th>Location</th>
```

```
                    <th>Date</th>
```

```
                </tr>
```

```
            </thead>
```

```
            <tbody>
```

```
        <?php
```

```
        while($row = $result->fetch_assoc()) {?
```

```
            <tr>
```

```
                <td><?php echo $row["id"]; ?></td>
```

```
                <td><?php echo $row["firstname"]; ?></td>
```

```
                <td><?php echo $row["lastname"]; ?></td>
```

```
                <td><?php echo $row["email"]; ?></td>
```

```
                <td><?php echo $row["age"]; ?></td>
```

```
                <td><?php echo $row["location"]; ?></td>
```

```
                <td><?php echo $row["date"]; ?> </td>
```

```

        </tr>
    <?php } ?>
</tbody>
</table>

<?php } else { ?>
    <blockquote>No results found for <?php echo escape($_POST['location
<?php
}

$conn->close();
}
?>

```

Am Schluss sollte Ihr read.php so aussehen:

```
<!doctype html>
<html lang="en">

<head>
    <meta charset="utf-8">
    <meta http-equiv="x-ua-compatible" content="ie=edge">
    <meta name="viewport" content="width=device-width, initial-scale=1">

    <title>Simple Database App</title>

    <link rel="stylesheet" href="css/style.css">
</head>

<body>
    <h1>Simple Database App</h1>

    <h2>Find user based on location</h2>

    <?php
        $servername = "localhost";
        $username = "root";
        $password = "password";
        $database = "usertest";

        // Create connection
        $conn = new mysqli($servername, $username, $password, $database);

        // Check connection
        if ($conn->connect_error) {
            die("Connection failed: " . $conn->connect_error);
        }
    ?>

    <form method="post">
```



```
<label for="location">Location</label>
<input type="text" id="location" name="location">
<input type="submit" name="submit" value="View Results">
</form>
```

```
<?php
```

```
if(isset($_POST["submit"])){
```

```
    $location = $_POST['location'];
```

```
    $sql = "SELECT * from users WHERE location='".$location.'";
```

```
    //echo $sql;
```

```
    $result = $conn->query($sql);
```

```
    if ($result->num_rows > 0) {?
```

```
        <h2>Results</h2>
```

```
        <table>
```

```
            <thead>
```

```
                <tr>
```

```
                    <th>#</th>
```

```
                    <th>First Name</th>
```

```
                    <th>Last Name</th>
```

```
                    <th>Email Address</th>
```

```
                    <th>Age</th>
```

```
                    <th>Location</th>
```

```
                    <th>Date</th>
```

```
                </tr>
```

```
            </thead>
```

```
            <tbody>
```

```
        <?php
```

```
        while($row = $result->fetch_assoc()) {?
```

```
            <tr>
```

```
                <td><?php echo $row["id"]; ?></td>
```

```
                <td><?php echo $row["firstname"]; ?></td>
```

```
                <td><?php echo $row["lastname"]; ?></td>
```

```

        <td><?php echo $row["email"]; ?></td>
        <td><?php echo $row["age"]; ?></td>
        <td><?php echo $row["location"]; ?><
        <td><?php echo $row["date"]; ?> </td>
    </tr>

    <?php } ?>

    </tbody>
</table>

    <?php } else { ?>
        <blockquote>No results found for <?php echo escape($_POST['location
    <?php
    }

    $conn->close();
}
?>

<a href="index.php">Back to home</a>

</body>
</html>

```

20.5 Abschluss

Die Applikation liesse sich jetzt natürlich beliebig erweitern...folgende Ideen:

- Darstellung anpassen, schöner machen
- Andere Filter einbauen
- Löschen von Datensätzen? (Delete-SQL-Befehl)
- Ändern von Datensätzen? (Update-SQL-Befehl)