

# LIFE EXPECTANCY ANALYZER

## PROJEKTBERICHT DBS HS2022

Team: “DankDaten”

**Oliver Grun** oliver.grun@stud.hslu.ch  
**Choekyel Nyungmartsang** choekyel.nyungmartsang@stud.hslu.ch  
**Tharrmeehan Krishnathasan** tharrmeehan.krishnathasan@stud.hslu.ch

## Management Summary

### Ausgangslage

Im Modul Datenbanksysteme erhielten wir die Aufgabe selbstständig in einer Gruppe an der Erstellung einer Datenbank inklusive aller Zwischenschritte zu arbeiten. Dabei mussten wir während des ganzen Semesters das dazu nötige Know-how erarbeiten und in unser Projekt implementieren. Es gab dabei 11 verschiedene Zwischenschritte, an denen wir uns orientieren konnten, von der Definition des Use-Cases für unser Projekt bis zur Visualisierung, um das Projekt durchzuführen.

### Vorgehen

Um den Zusammenhang zwischen der Verfügbarkeit von Trinkwasser und der Lebenserwartung in verschiedenen Ländern analysieren zu können, haben wir Daten der World Health Organization (von nun an: WHO) mit Daten verschiedener Non-Profit-Organisationen (von nun an: NPO) abgeglichen. Die für uns irrelevanten und fehlerhaften Daten haben wir herausgelöscht und die relevanten Daten miteinander verknüpft, sodass wir einen Zusammenhang eruieren konnten. Die Visualisierung erfolgte mithilfe einer Streamlit-Applikation, mit der Interessierte unsere Auswertungen anschauen können.

### Ergebnis

Das Ergebnis zeigte, dass eine positive Korrelation zwischen der Verfügbarkeit von Trinkwasser und Lebenserwartung vorhanden ist. Dies lässt sich damit begründen, dass die Grundbedürfnisse von den Ländern mit einer höheren Verfügbarkeit fast zu 100% gedeckt sind, sodass sie sich auf andere Dinge konzentrieren, welche die Lebenserwartung erhöhen können. Ärmere Länder müssen ihre Grundbedürfnisse, wie Trinkwasser und Nahrung, decken weshalb sie sich nicht auf die Erhöhung der Lebenserwartung konzentrieren können. Mehrfach spielen auch Faktoren wie Krieg, Katastrophen und andere ähnliche Probleme eine Rolle, welche die Lebenserwartung temporär sinken lassen, da vermehrt Menschen jeder Altersgruppe sterben.

<b><u>1</u></b>	<b><u>PROJEKTAUFRAG</u></b>	<b>4</b>
<b><u>2</u></b>	<b><u>USE CASE</u></b>	<b>4</b>
<b>2.1</b>	<b>PROBLEMSTELLUNG</b>	<b>4</b>
<b>2.2</b>	<b>ZIELSETZUNG</b>	<b>4</b>
<b><u>3</u></b>	<b><u>AUSWAHL DER DATEN / DATENGRUNDLAGE</u></b>	<b>5</b>
<b>3.1</b>	<b>LEBENSERWARTUNG</b>	<b>5</b>
<b>3.2</b>	<b>TRINKWASSERZUGANG</b>	<b>6</b>
<b>3.3</b>	<b>SÄUGLINGSSTERBLICHKEIT</b>	<b>6</b>
<b>3.4</b>	<b>NON-PROFIT-ORGANISATIONEN</b>	<b>7</b>
<b><u>4</u></b>	<b><u>SYSTEMARCHITEKTUR</u></b>	<b>8</b>
<b><u>5</u></b>	<b><u>DATENBANK</u></b>	<b>9</b>
<b>5.1</b>	<b>DATENBANK TECHNOLOGIE</b>	<b>9</b>
<b>5.2</b>	<b>DATENMODELL (ENTITY-RELATIONSHIP-DIAGRAMM)</b>	<b>9</b>
<b>5.3</b>	<b>TABELLENSTRUKTUR (ERD-MODELLIERUNG)</b>	<b>10</b>
<b>5.4</b>	<b>SQL-SKRIPT / SCHEMA</b>	<b>11</b>
<b><u>6</u></b>	<b><u>DATENMIGRATION / DATENIMPORT</u></b>	<b>12</b>
<b>6.1</b>	<b>DATEN BEREINIGUNG / DATENAUFBEREITUNG</b>	<b>12</b>
<b>6.2</b>	<b>DATENIMPORT IN DATENBANK</b>	<b>12</b>
<b><u>7</u></b>	<b><u>PERFORMANCE</u></b>	<b>13</b>
<b>7.1</b>	<b>DENORMALISIEREN</b>	<b>14</b>
<b><u>8</u></b>	<b><u>DATENBANK AUSWERTUNG</u></b>	<b>15</b>
<b>8.1</b>	<b>VIEW QUERIES</b>	<b>15</b>
<b>8.2</b>	<b>ARCHITEKTUR</b>	<b>15</b>
<b><u>9</u></b>	<b><u>DATENBANKABFRAGEN</u></b>	<b>16</b>
<b><u>10</u></b>	<b><u>VISUALISIERUNGEN</u></b>	<b>17</b>
<b>10.1</b>	<b>VORGEHEN DER VISUALISIERUNG</b>	<b>17</b>
<b><u>11</u></b>	<b><u>SICHERHEITSASPEKTE</u></b>	<b>19</b>
<b>11.1</b>	<b>DATENBANK SERVER</b>	<b>19</b>
<b>11.2</b>	<b>AUTHENTISIERUNG UND AUTORISIERUNG</b>	<b>19</b>

<b>11.3</b>	<b>BENUTZER</b>	<b>20</b>
<b>11.4</b>	<b>ZUSÄTZLICHE SICHERHEIT</b>	<b>21</b>
<b>11.5</b>	<b>RISIKEN</b>	<b>22</b>
<b>11.6</b>	<b>AUDITING</b>	<b>22</b>
<b><u>12</u></b>	<b>TESTS</b>	<b><u>23</u></b>
<b>12.1</b>	<b>FUNKTIONALE TESTS</b>	<b>23</b>
<b>12.2</b>	<b>PERFORMANCETEST</b>	<b>23</b>
<b>12.3</b>	<b>SICHERHEITSTESTS</b>	<b>24</b>
<b><u>13</u></b>	<b>ERGEBNIS</b>	<b><u>25</u></b>
<b><u>14</u></b>	<b>REFLEXION</b>	<b><u>26</u></b>
<b>14.1</b>	<b>CHOEKYEL N.</b>	<b>27</b>
<b>14.2</b>	<b>OLIVER G.</b>	<b>27</b>
<b>14.3</b>	<b>THARRMEEHAN K.</b>	<b>27</b>
<b><u>15</u></b>	<b>LITERATURVERZEICHNIS</b>	<b><u>28</u></b>
<b><u>16</u></b>	<b>ABBILDUNGSVERZEICHNIS</b>	<b><u>28</u></b>
<b><u>17</u></b>	<b>ANHANG</b>	<b><u>29</u></b>
<b>17.1</b>	<b>CREATE &amp; LOAD QUERIES</b>	<b>29</b>
<b>17.2</b>	<b>VIEW QUERIES</b>	<b>31</b>
<b>17.3</b>	<b>PASSWORT PLUGIN QUERY</b>	<b>32</b>
<b>17.4</b>	<b>USER QUERIES</b>	<b>32</b>
<b>17.5</b>	<b>VISUALISIERUNGEN</b>	<b>33</b>
<b>17.6</b>	<b>INDEX QUERY</b>	<b>34</b>
<b><u>18</u></b>	<b>PROJEKTPLAN</b>	<b><u>35</u></b>

## 1 Projektauftrag

Der nachfolgende technische Bericht ist Bestandteil der Leistungserbringung des Moduls Datenbankstrukturen der Hochschule Luzern, Departement Informatik. Ziel dieser Arbeit ist es, die erlernten Inhalte des Moduls zu vertiefen und praktisch anzuwenden.

Das Projekt wurde im Verlauf des gesamten Herbstsemesters 2022 in einem Team von 3 Personen selbstständig erarbeitet. Das Projektergebnis soll einen Entscheidungsträger in einem beispielhaften Anwendungsfall unterstützen, indem es Informationen liefert, die durch eine Datenbankabfrage oder -analyse gewonnen wurden.

## 2 Use Case

### 2.1 Problemstellung

Heutzutage gibt es immer grosse Unterschiede der Lebenserwartung in den einzelnen Ländern. Es gab schon viele Studien in der Vergangenheit zu den einzelnen Faktoren, welche die Lebenserwartung in einem Land beeinflussen. Wir beziehen uns in dieser Arbeit auf den Faktor, der Zugang zu sauberem Trinkwasser in einem Land und ob dieser helfen kann, die Lebenserwartung vom jeweiligen Land zu erhöhen.

Die Lebenserwartung bezieht sich auf die Anzahl der Jahre, die eine Person im statistischen Durchschnitt voraussichtlich leben wird. Sie hängt von verschiedenen Faktoren eines Landes ab. Vor der Modernisierung der Welt lag die Lebenserwartung in allen Teilen der Welt bei etwa 30 Jahren. Zu Beginn des 19. Jahrhunderts stieg die Lebenserwartung in den industriellen Ländern stark an, während sie in den anderen Ländern nur leicht anstieg. Heutzutage liegt die Lebenserwartung auf der ganzen Welt bei 72 Jahren, während in Afrika die Lebenserwartung bei 63 Jahren liegt.<sup>1</sup>

In den Ländern, bei denen die Lebenserwartung niedriger ist, ist auch meist die Kindersterblichkeitsrate deutlich höher. Dies hat auch verschiedene Faktoren, wobei wir uns auch hier auf die Verfügbarkeit von Trinkwasser fokussieren.

### 2.2 Zielsetzung

Das Ziel dieses Projektes ist es, mit Hilfe von klaren Visualisierungen aufzuzeigen, welche positive Einflüsse der Zugang zu Trinkwasser auf die Lebenserwartung und auf die Kindersterblichkeitsrate hat. Durch diese visuellen Analysen soll es den NPOs ermöglichen, einfacher und klarer eine Kampagne in einem Land zu erstellen und Geldspenden dafür zu bekommen. Durch die Visualisierungen können NPOs auch herausfinden, welche Länder noch am dringendsten geholfen werden muss.

Somit soll dieses Projekt indirekt helfen, die Wasserversorgung in ärmeren Ländern, welche Hilfe gebrauchen zu optimieren und damit auch die Lebenserwartung, sowie die Kindersterblichkeitsrate auf der ganzen Welt zu verbessern.

---

<sup>1</sup> (Rudnicka, Statistiken zum Thema Lebenserwartung, 2022)

### 3 Auswahl der Daten / Datengrundlage

Für die Analyse ob ein Zusammenhang zwischen der Lebenserwartung, der Verfügbarkeit von Trinkwasser und der Kindersterblichkeitsrate besteht, mussten wir zuerst die nötigen Daten bekommen. Diese Daten konnten wir auf [www.kaggle.com](https://www.kaggle.com) finden. Es ist ein vorgefertigtes Datenset der WHO. Die Daten stammen aus dem Datenarchiv des globalen Gesundheitsobservatoriums (GHO) der WHO. Sie verfolgen den Gesundheitszustand sowie viele andere damit zusammenhängende Faktoren für alle Länder. Das CSV-File besteht aus knapp 3'000 Datensätzen und 8 Spaltenüberschriften (Zeile 1). Die einzelnen Attribute sind außerdem durch ein Komma getrennt. Für die Visualisierungen, welche Länder von NPO unterstützt werden, haben wir die Daten von den Top 5 NPOs, welche sich auf die Wasserversorgung in einem Land fokussieren genommen. Damit wir die Daten auf eine Weltkarte visualisieren konnten, brauchten wir zusätzlich noch das Datenset von Natural Earth. Dieses Datenset beinhaltete auch noch Informationen zu der Bevölkerungsgröße und der unterschiedlichen Schreibarten der Ländernamen. Die Datensätze, die wir für die Analyse verwendet haben, sind in der nachfolgenden Tabelle ersichtlich.

Bezeichnung	Grösse	Format	Quelle
lifeExpectancyAtBirth.csv	139.87 kB	csv	<a href="#">Link</a>
basicDrinkingWaterServices.csv	285.48 kB	csv	<a href="#">Link</a>
infantMortalityRate.csv	4'000 kB	csv	<a href="#">Link</a>
Natural Earth Countries	4'700 kB	shp	<a href="#">Link</a>

#### 3.1 Lebenserwartung

Durch die Abbildung sieht man, dass die meisten Länder mit einer niedrigen Lebenserwartung in Asien und in Afrika sind. Das Land mit der niedrigsten Lebenserwartung liegt jedoch in Europa und ist Lettland mit 48.4 Jahren.

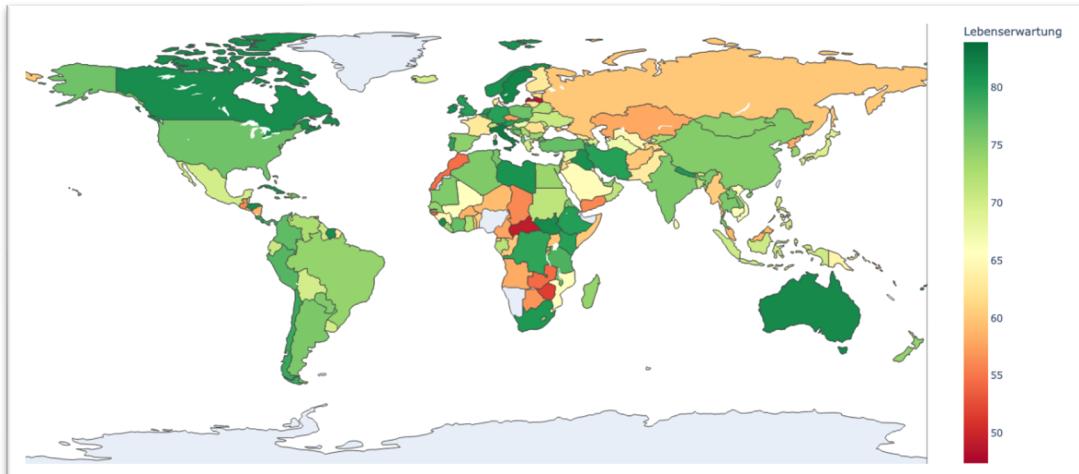


Abbildung 1: Lebenserwartung von [1920, 2000, 2010, 2015, 2019] (Quelle: Eigene Anfertigung)

### 3.2 Trinkwasserzugang

Durch die Karte des prozentualen Anteils des Trinkwasserzugangs eines Landes kann man sehen, welche Länder guten Zugang und welche Länder schlechten Zugang zu Trinkwasser haben. Auch hier sieht man, dass die meisten Länder mit einem schlechten Zugang zu Trinkwasser in Asien und in Afrika sind. Wenn man die nachfolgende Karte mit der vorherigen vergleicht, sieht man, dass gewisse Länder mit schlechtem Trinkwasserzugang auch eine schlechte Lebenserwartung haben.

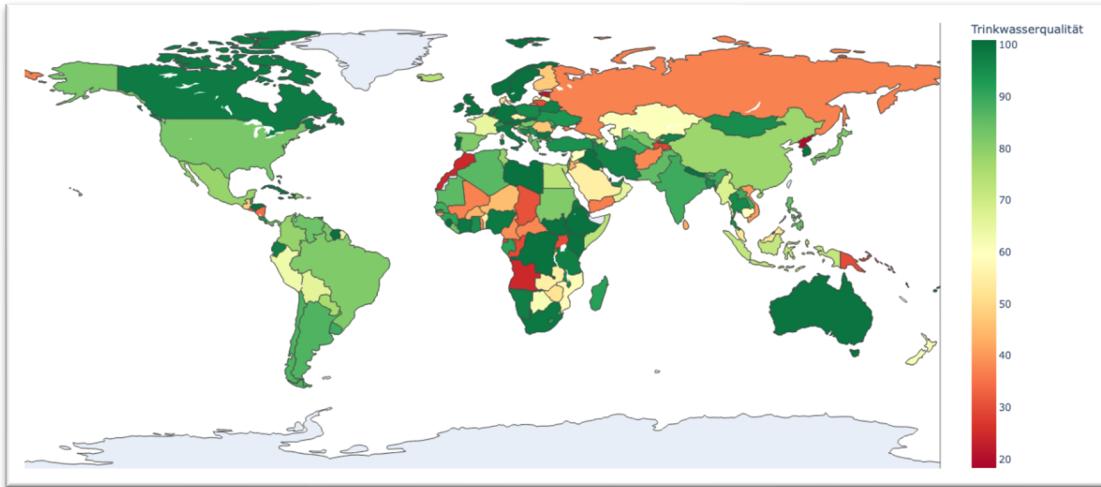


Abbildung 2: Trinkwasserzugang von [2000-2017] (Quelle: Eigene Anfertigung)

### 3.3 Säuglingssterblichkeit

Die Säuglingssterblichkeitsrate auf dieser Karte wird in Fällen pro 1000 Geburten angezeigt. Auch hier sieht man wieder, dass Länder mit schlechtem Trinkwasser Zugang eine hohe Säuglingssterblichkeitsrate haben. Hier muss man jedoch beachten, dass nicht alle Länder diese Daten erheben und veröffentlichen. Ein Säuglingstodesfall wird definiert, wenn ein Kind zwischen der Geburt und seinem ersten Lebensjahr stirbt.

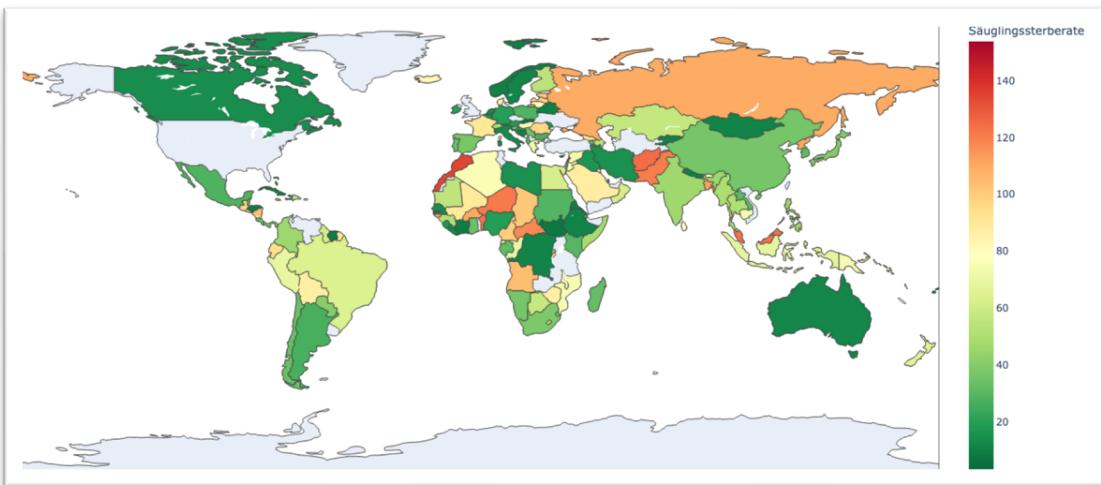


Abbildung 3: Säuglingssterblichkeit von [1950-2019] (Quelle: Eigene Anfertigung)

### 3.4 Non-Profit-Organisationen

Die 5 NPOs wurden von uns händisch herausgesucht. Wir haben uns dabei auf grosse Internationale NPOs bezogen, welche den Trinkwasser Zugang für Länder erhöhen wollen. Nach unseren Recherchen sind wir auf diese fünf Organisationen gestossen.

Organisation	Webseite
Splash	<a href="#">Link</a>
Blood:Water	<a href="#">Link</a>
Pure Water: For the World	<a href="#">Link</a>
Water for Good	<a href="#">Link</a>
Lifewater	<a href="#">Link</a>



Abbildung 4: 5 Non-Profit-Organisationen (Quelle Eigene Anfertigung)

## 4 Systemarchitektur

Um die Entscheidungsunterstützung zu realisieren, wird eine Datenbank mit dazugehörigem Datenbankmanagementsystem benötigt. Ein solches Konstrukt lässt sich auf verschiedene Arten realisieren. Um die volle Kontrolle über die Datenbank und die darin enthaltenen Datensätze zu erhalten, haben wir uns dazu entschieden, die Datenbank auf einer Virtuellen Maschine in der Enterprise Lab Umgebung laufen zu lassen. Die virtuelle Umgebung ist mit einem Windows Server 2022 (Datacenter Edition) ausgestattet. Auf dem Server haben wir einen MySQL Datenbank Server eingerichtet. Diese Systemarchitektur ermöglicht es uns parallel mit dem gleichen Datenstand zu arbeiten, unabhängig von Standort der Entwickler. Via VPN der Hochschule Luzern greifen wir auf die Enterprise Lab Umgebung zu, in welcher unsere Datenbank installiert ist.

Für die Entwicklung des Entscheidungsunterstützungssystems wird hauptsächlich die Workbench von MySQL benutzt, der von Oracle entwickelt wurde. Dieses Tool bietet viele Funktionen, vom Erstellen von ERM-Diagrammen, bis zum Prototyping. Die Auswertung der Datenbank visualisieren wir mit Python und geben diese auf einem Dashboard aus.

Die folgende Visualisierung zeigt die Systemarchitektur auf:

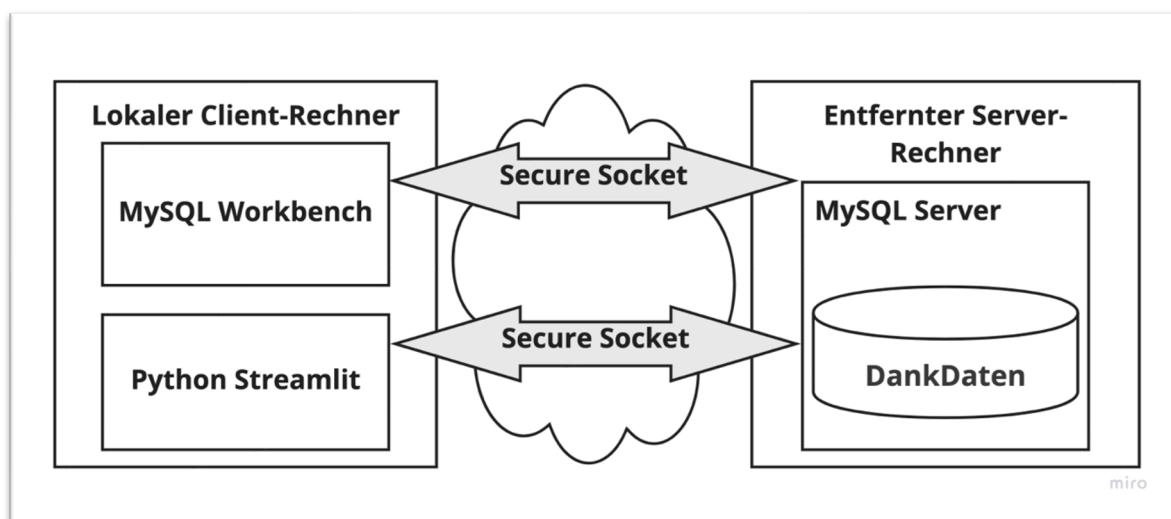


Abbildung 5: Modell Systemarchitektur (Quelle: Eigene Anfertigung)

## 5 Datenbank

Dieses Kapitel gibt einen detaillierten Überblick über die für das Projekt aufgebaute Datenbank. Es wird beschrieben welche Datenbank aus welchem Grund verwendet wird, wie diese Datenbank aufgebaut ist und wie die Daten aufbereitet und importiert wurden.

### 5.1 Datenbank Technologie

Bei der Technologie der Datenbank haben wir uns für eine relationale Datenbank entschieden. Wir benutzen in unserem Projekt klar strukturierte Daten, welche offensichtliche Relationen zueinander haben. Dadurch, dass wir verschiedene Abfragen für die Webapplikation machen mussten, haben wir entschieden, dass dies für uns am einfachsten geht, wenn wir SQL verwenden. Weil wir schon gewisse Erfahrungen mit MySQL hatten, beschlossen wir uns die Datenbank in MySQL zu erstellen und zu gebrauchen.

Wir haben uns bewusst gegen eine Graphdatenbank entschieden, da wir die Beziehung gut in Tabellenform speichern konnten, auch muss unsere Datenbank nicht allzu skalierbar sein, da unsere Daten konstant sind oder es nur sehr wenige Veränderungen gibt. Die Dokumentendatenbank ist für unsere Datenstruktur sehr unpassend und wir hätten viele Beziehungen denormalisieren müssen, deswegen haben wir uns auch nicht für eine Dokumentdatenbank entschieden.

### 5.2 Datenmodell (Entity-Relationship-Diagramm)

Unser Entity-Relationship-Diagramm (von nun an: ERD) umfasst sechs Entitäten: LifeExpectancy (Lebenserwartung), InfantDeathRate (Kindersterblichkeitsrate), DrinkingWaterSupply (Trinkwasserverfügbarkeit), Continent (Kontinent), Country (Land), Organisation (Organisation), sowie eine Beziehungstabelle: country\_organisation (Land und Organisation). Zu Beginn haben wir versucht, die Daten, welche wir für unseren Use Case benötigen, mit einer Skizze auf einem ERD aufzuzeichnen. Nachdem wir wussten, was wir für Daten speichern wollen, haben wir uns um die Normalisierung in die dritte Normalform gekümmert. Im ERD sieht man die einzelnen Relationen sehr gut, zum Beispiel sieht man, dass ein Land nur auf einem Kontinent sein kann, aber ein Kontinent mehrere Länder besitzt (1:n). Dieses ERM war unsere Vorlage für die Erstellung der Datenbank.

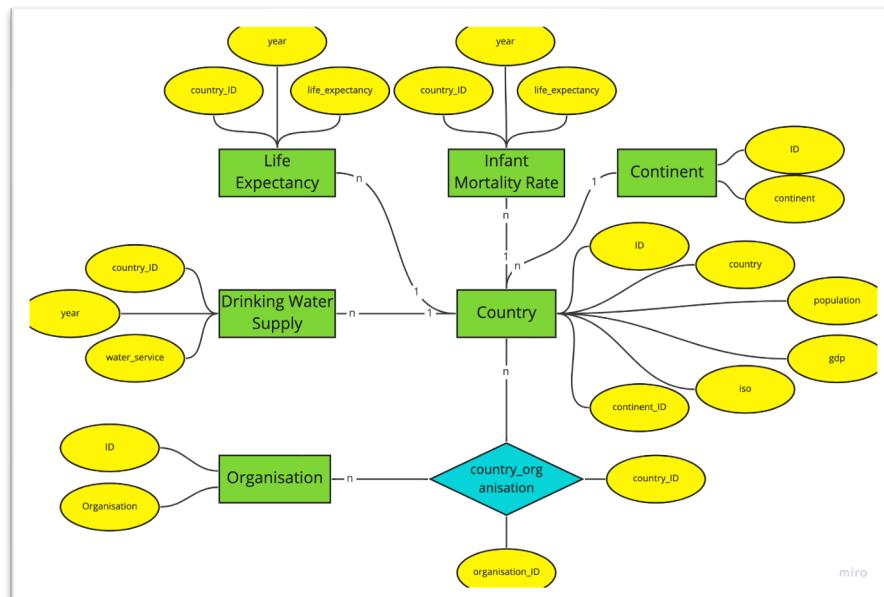


Abbildung 6: Entity-Relationship-Diagramm (Quelle: Eigene Anfertigung)

### 5.3 Tabellenstruktur (ERD-Modellierung)

Anhand des Entity-Relationship-Diagramm (ERD) wurden die Tabellen in der MySQL Workbench erstellt. Im Vergleich zum ERD ist die Tabellenstruktur in der Workbench umfassender: Zu jeder Tabelle gehören eine Vielzahl von Attributen.

Wir haben explizit darauf geachtet Primary- und Foreign Keys zu erstellen, um Konsistenz unserer Daten sicherzustellen mittels referenzieller Integrität. Die definierten Schlüsselkandidaten sind aus der Abbildung zu entnehmen.

Die Modellierung wurde in der MySQL Workbench erstellt. Mit Hilfe von diesem kann im Falle eines Ausfalls der Datenbank neu generiert werden.

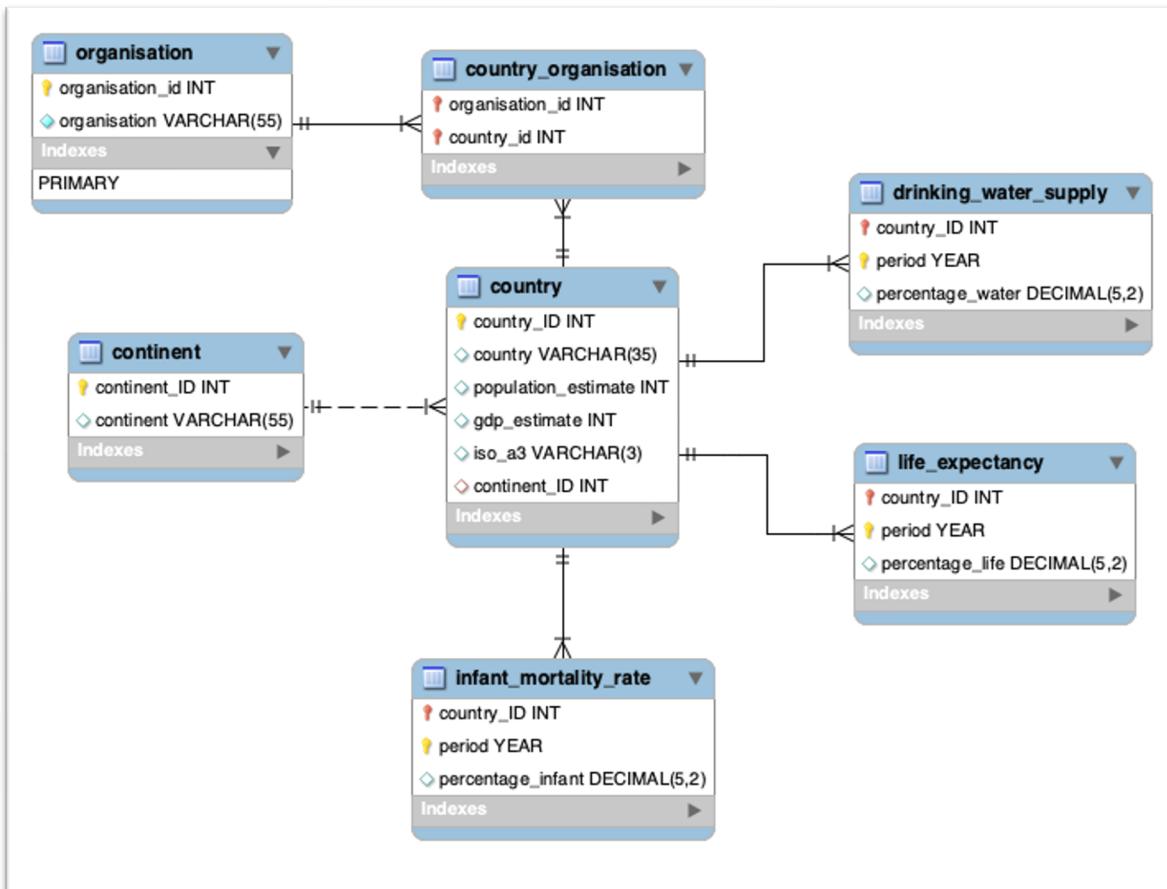


Abbildung 7: Entity-Relationship-Modell der Datenbank (Quelle: Eigene Anfertigung)

## 5.4 SQL-Skript / Schema

Nach der Überprüfung des Entity-Relationship-Diagramm, ging es daran die Tabellen und Attribute zu erstellen. Das Schema wurde als SQL-File abgespeichert, um auch hier die Wiederherstellung zu garantieren oder das Schema zu replizieren.

Nachfolgend sieht man einen kleinen Auszug aus dem SQL-Skript. Der vollständige Query ist unter «CREATE & LOAD Queries» zu finden.



```
1 -----  
2 -- create drinking_water_supply table  
3 -----  
4 CREATE TABLE drinking_water_supply  
5 (  
6 country_ID INT,  
7 period YEAR,  
8 percentage_water DECIMAL(5,2),  
9 PRIMARY KEY (country_ID, period),  
10 FOREIGN KEY (country_ID) REFERENCES country(country_ID);  
11 -----  
12 -- load drinking water supply csv in table  
13 -----  
14 LOAD DATA LOCAL INFILE '/Users/oliver/HSLU/DBS/csv/transform/drinking_water_supply.csv'  
15 INTO TABLE drinking_water_supply FIELDS TERMINATED BY ',' OPTIONALLY ENCLOSED  
16 BY '\"' ESCAPED BY '\\\' LINES TERMINATED BY '\n'  
17 IGNORE 1 LINES (country_ID, period, percentage_water);
```

Abbildung 8: Auszug vom CREATE TABLE (Quelle: Eigene Anfertigung)

## 6 Datenmigration / Datenimport

Um die Daten, welche im Kapitel 3 Datengrundlage beschrieben wurden, in unsere Datenbank zu importieren war vorgängig noch eine Datenaufbereitung notwendig. Der genaue Prozess von den Rohdaten bis in die Datenbank wird in diesem Abschnitt beschrieben. Den Prozess unseres Datenimports stimmt mit dem bereits bekannten Extract, Transform, Load (ETL)<sup>2 3</sup> Prozess überein.

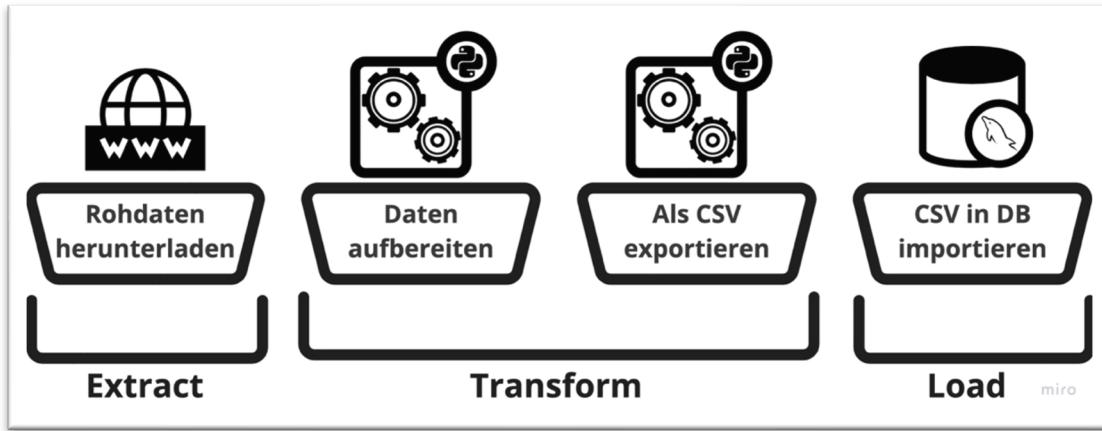


Abbildung 9: Prozess von den Rohdaten bis in unsere Datenbank (Quelle: Eigene Anfertigung)

### 6.1 Daten Bereinigung / Datenaufbereitung

Die Aufbereitung der Daten erfolgt mittels Python. Wir haben uns entschieden die Datenaufbereitung mit Python durchzuführen, da wir deutlich mehr Möglichkeiten haben die Daten auf unsere Wünsche und Bedürfnisse anzupassen als mit SQL. Zu unserer Entscheidung beigetragen hat auch die Tatsache, dass Python mittlerweile das Werkzeug ist, wenn es um Datentransformationen geht.<sup>4 5</sup> Zudem können wir mit diesen Skripts im Nachgang, auf einfache Art und Weise, weitere Anpassungen vornehmen und diese wieder in unsere Datenbank importieren.

In den Skripts werden die Daten in verschiedenen CSV-Files kostenlos von [www.kaggle.com](https://www.kaggle.com) (Kaggle, 2022) heruntergeladen [Extract]. Danach transformieren wir die Daten, sodass sie zu unserem definierten Schema passen. Das heisst wir löschen Attribute, die wir nicht brauchen, benennen die übrigen Attribute entsprechend um und bringen die Daten in die gewünschte Struktur [Transform]. Zuletzt exportieren wir die bereinigten Daten in verschiedene CSV-Files. Für jede Tabelle des Schemas gibt es ein CSV-File, welches importiert werden kann [Load]. Das vollständige Skript ist im Anhang verfügbar unter «CREATE & LOAD Queries».

### 6.2 Datenimport in Datenbank

Für den Import der generierten CSV-Daten haben wir zwei Wege getestet: Über den Integrierten Wizard in der MySQL-Workbench und direkt über ein SQL-Skript, welches die generierten CSVs einliest.

Die zweite Variante mit dem Einlesen des Skripts hat für uns den grössten Mehrwert erbracht. Wir konnten die Parameter der einzelnen Spalten direkt angeben, wobei damit die Geschwindigkeit positiv beeinflusst wird und die Versionskontrolle konnte einfacher gemacht werden. Für grosse Anzahl an Daten ist es wichtig, dass die Geschwindigkeit Optimierung von Anfang an in Betracht zu ziehen.

<sup>2</sup> (Rudnicka, 2022)

<sup>3</sup> (Bansal & Kagemann, 2015)

<sup>4</sup> (Sanyal, 2022)

<sup>5</sup> (Jalli)

## 7 Performance

Um die Leistung einer Datenbank generell und in unserem spezifischen Fall einer SQL-Datenbank zu verbessern, gibt es verschiedene Möglichkeiten. Auf jeden Fall lohnt es sich, einen Blick auf den Ausführungsplan zu werfen. Nachfolgend zu sehen ist eine SQL Query, wie sie in unserem Use Case verwendet wird:

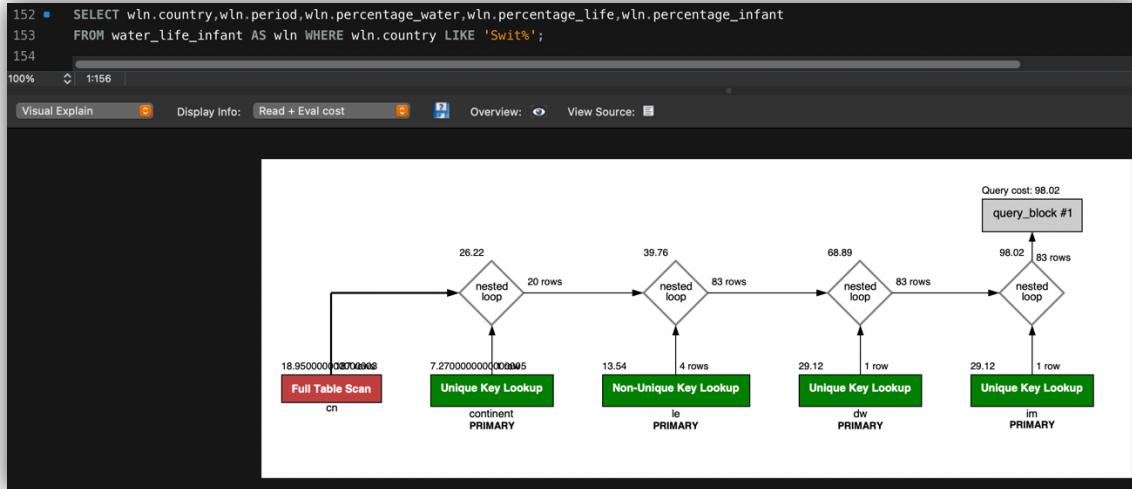


Abbildung 10: Select Switzerland ohne Indexierung (Quelle: Eigene Anfertigung)

Im dazugehörigen Ausführungsplan ist gut zu sehen, dass für die Ausführung der Query bereits die "Primary Keys" verwendet werden können. Das macht die Ausführung sehr performant. Lediglich der Zugriff auf die Reihe "Switzerland" kann nicht von einem Index profitieren. Die obige Query benötigt 0.023 Sekunden und hat eine Query cost von 98.02.

Nun könnte man davon ausgehen, dass es Sinn macht, auf "country(country\_ID)" einen Unique Index zu erstellen.

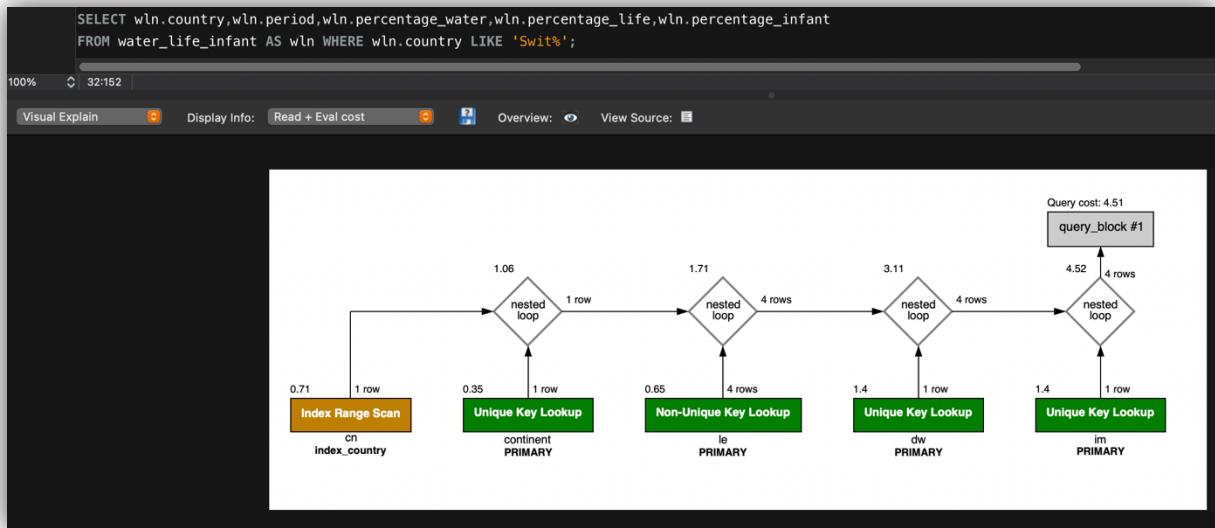


Abbildung 11: Select Switzerland mit Indexierung (Quelle: Eigene Anfertigung)

Eine erneute Einsicht in den Ausführungsplan zeigt, dass die Query nun von dem neu erstellten Unique Index auf "country(country\_ID)" profitiert. Die Ausführung dauert nun 0.014 Sekunden und hat eine Query cost von 4.51. Eine kleine Verbesserung, aber eine Verbesserung nichtsdestotrotz.

Dieses Resultat ist grundsätzlich einfach zu erklären. Die Tabelle "country" enthält 187 Einträge. Es ist daher für die Datenbank eine einfache Aufgabe, diese Einträge zu durchsuchen, auch ohne Index.

Es gibt jedoch auch andere Mittel zur Verbesserung der Performance. Es macht Sinn, die Datentypen so zu wählen, dass sie möglichst wenig Speicher benötigen. So ist es zum Beispiel ideal, bei kleinen Indizes den Datentypen "TINY INT" zu wählen. Dieses Vorgehen kann man über die gesamte Datenbankstruktur weiterführen, sprich Datentypen so weit einschränken, dass sie nicht unnötig gross sind. Daher ist es sehr wichtig, bereits bei der ERM-Modellierung genau zu wissen, mit welchen Daten man arbeitet.

Ein anderer Ansatz zur Leistungssteigerung wäre die Parallelisierung. Dieses Konzept macht jedoch nur dann Sinn, wenn enorme Datenmengen verarbeitet werden müssen. Ein Beispiel aus der realen Welt ist die Monats-end-Verarbeitung einer Bank. Es müssen für unzählige Menschen Kontoauszüge etc. erstellt werden. In diesem Fall kann die Parallelisierung zu einem enormen Performancezuwachs führen. In unserem Fall macht ein solches Vorgehen jedoch wenig Sinn.

Zusammengefasst kann gesagt werden, dass in diesem Projekt keine Massnahmen zur Optimierung der Performance getroffen werden müssen. Die Datenmengen, welche verarbeitet werden, sind schlichtweg zu klein. Die Grösste Tabelle "infant\_mortality\_rate" enthält 1'000 Einträge. Das ist für ein modernes Datenbanksystem keine riesige Herausforderung. Das macht es uns auch möglich, Funktionen wie "LIKE" zu benutzen, ohne dass der Endbenutzer lange auf die Auswertung warten muss.

## 7.1 Denormalisieren

Wir haben uns entschieden, das Jahr in den einzelnen Tabellen zu lassen, da die meisten Abfragen mit Hilfe der Jahreszahl gemacht werden. Hierbei wäre ein Performance Verlust vorhanden gewesen, wenn wir das Jahr in eine eigene Tabelle abgespeichert hätten, da danach mit «JOIN» Befehl die Werte aus der Jahrestabelle abgelesen hätten müssen.

Da wir mit den gesetzten Indices eine Performanceverbesserung erreicht hatten, haben wir uns dazu entschieden nicht noch weitere Denormalisierungen in der Datenbank vorzunehmen. Wir sind mit der Performance von weniger als einer Sekunde sehr zufrieden.

## 8 Datenbank Auswertung

In diesem Kapitel wird beschrieben, wie wir die Auswertung unserer Daten vollzogen haben und welche Visualisierung daraus entstanden sind.

### 8.1 VIEW Queries

Da wir für die Auswertung der Visualisierung, verschiedenste Informationen benötigen, mussten wir zwangshalber mehrheitlich alle verwendeten Tabellen mit der «JOIN»-Funktion verbinden. Somit haben wir rund acht Tabellen mit der Tabelle "country" zusammengeführt, wodurch wir Zugriff auf sämtliche Attribute für eine optimale Ausgabe erhalten. Das vollständige File ist im Anhang ersichtlich unter «VIEW Queries».

### 8.2 Architektur

Für das Erstellen der Visualisierungen haben wir uns entschieden mit der Programmiersprache Python zu arbeiten, da wir uns in dieser Sprache alle wohlfühlen. Mittels Python greifen wir über einen MySQL Connector auf die erstellte Datenbank zu. Infolgedessen müssen wir mit Python keine weiteren Datentransformation oder Datenaggregationen vornehmen, sondern können die aufbereiteten Daten der Views direkt verwenden. Ferner gibt uns das Arbeiten mit Views die Möglichkeit von den mächtigen SQL SELECT Optionen zu profitieren. Für den Zugriff auf die Views gibt es einen speziell dafür angelegten Benutzer, welcher nur Abfrage Zugriffe auf die Datenbank hat.

Die Grafiken haben wir zur Übersicht in *Streamlit*<sup>6</sup> mit einem Dashboard zusammengefasst und visualisiert. *Streamlit* ist ein Python Package, um schnell und einfach Interaktive Visualisierungen, sowie einfache Dashboards zu erstellen. So entsteht ein Tool, welches hilft, den Zusammenhang von der Lebenserwartung, der Verfügbarkeit von Trinkwasser und der Säuglingssterblichkeit besser zu verstehen.

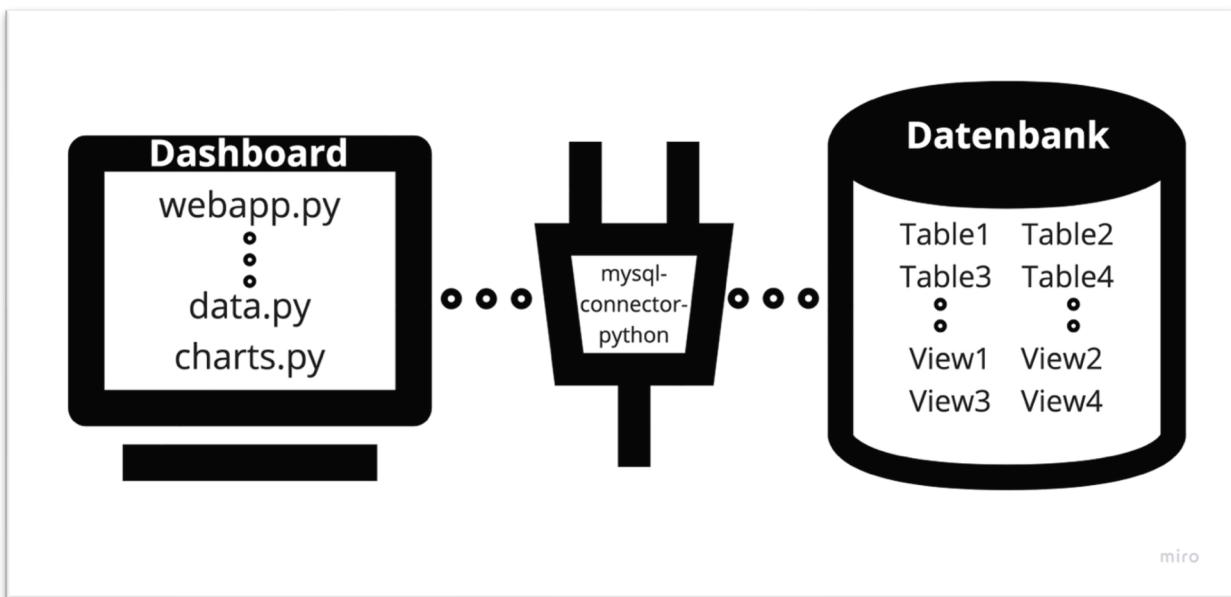


Abbildung 12: Architekturübersicht unserer Datenvisualisierung (Quelle: Eigene Anfertigung)

<sup>6</sup> (Luber & Litzel, 2020)

## 9 Datenbankabfragen

Für die Datenbankabfragen arbeiten wir mit Views: In der Form eines Starschema beziehen diese virtuellen Tabellen einzelne Attribute aus den bestehenden Tabellen. Die Views dienen als Schnittstelle zum Dashboard, in welchem wir die aggregierten Daten visualisieren.

Wir haben verschiedene Views erstellt, um unsere Datenabfrage mit Hilfe von Python einfacherer und sicherer zu gestalten. Die Views ermöglichen uns die komplizierteren «SELECT»-Queries auf der Datenbank zu definieren und diese dann abzurufen, anstatt die SELECT-Statements mehrmals in Python neu zu schreiben.

```

1 • CREATE |
2     ALGORITHM = UNDEFINED
3     DEFINER = CURRENT_USER()
4     SQL SECURITY DEFINER
5     VIEW `choose_switzerland` AS
6         SELECT
7             `wln`.`population_estimate` AS `population_estimate`,
8             `wln`.`continent` AS `continent`,
9             `wln`.`iso_a3` AS `iso_a3`,
10            `wln`.`gdp_estimate` AS `gdp_estimate`,
11            `wln`.`country` AS `country`,
12            `wln`.`period` AS `period`,
13            `wln`.`percentage_water` AS `percentage_water`,
14            `wln`.`percentage_life` AS `percentage_life`,
15            `wln`.`percentage_infant` AS `percentage_infant`
16        FROM
17            `water_life_infant` `wln`
18        WHERE
19            (`wln`.`country` = 'Switzerland')

```

Abbildung 13: View für die Daten der Schweiz (Quelle: Eigene Anfertigung)

```

1 • CREATE
2     ALGORITHM = UNDEFINED
3     DEFINER = CURRENT_USER()
4     SQL SECURITY DEFINER
5     VIEW `water_life_infant` AS
6         SELECT
7             `cn`.`population_estimate` AS `population_estimate`,
8             `continent`.`continent` AS `continent`,
9             `cn`.`iso_a3` AS `iso_a3`,
10            `cn`.`gdp_estimate` AS `gdp_estimate`,
11            `cn`.`country` AS `country`,
12            `dw`.`period` AS `period`,
13            `dw`.`percentage_water` AS `percentage_water`,
14            `le`.`percentage_life` AS `percentage_life`,
15            `im`.`percentage_infant` AS `percentage_infant`
16        FROM
17            (((`country` `cn`
18            LEFT JOIN `continent` ON ((`cn`.`continent_ID` = `continent`.`continent_ID`)))
19            LEFT JOIN `drinking_water_supply` `dw` ON ((`cn`.`country_ID` = `dw`.`country_ID`)))
20            LEFT JOIN `infant_mortality_rate` `im` ON (((`dw`.`country_ID` = `im`.`country_ID`)
21                AND (`dw`.`period` = `im`.`period`))))
22            JOIN `life_expectancy` `le` ON (((`im`.`country_ID` = `le`.`country_ID`)
23                AND (`im`.`period` = `le`.`period'))))

```

Abbildung 14: View für die Statistiken jedes Landes (Quelle: Eigene Anfertigung)

Da wir die Funktionalitäten bei unserer Applikation mit Views lösen konnten und keine weiteren Datenmanipulationen auf der Datenbank vornehmen, haben wir uns beschlossen keine «Stored Procedures» zu erstellen. Falls wir in Zukunft unsere Webapplikationen erweitern würden und neue Daten auf der Datenbank zu speichern, würde es sich natürlich lohnen dafür ein «Stored Procedure» zu erstellen, damit wir die Sicherheit und die Performance dieses Tasks optimieren würden.

## 10 Visualisierungen

Für das Verständnis unserer Daten sind Visualisierungen zentral. In diesem Kapitel dokumentieren wir die wichtigsten Grafiken.

### 10.1 Vorgehen der Visualisierung

Unser Vorgehen war von Anfang an systematisch: Zuerst haben wir die Lebenserwartung mit der Verfügbarkeit von Trinkwasser im Zusammenhang gestellt und das geplottet. Anhand des Scatterplots konnte die Korrelation zwischen den Beiden sichtbar gemacht werden. In einem zweiten Schritt wurde die Lebenserwartung mit der Säuglingssterblichkeit im Zusammenhang gestellt. Dadurch konnten wir sehen, dass diese ebenfalls eine Korrelation aufweist. Als drittes haben wir dann die Säuglingssterblichkeit in Zusammenhang mit dem Trinkwasser Zugang visualisiert. Durch verschiedene Meetings sind wir dann darauf gekommen, welche Plots wir auf der Webseite anzeigen lassen wollen. Wir haben uns entschlossen, die verschiedenen Faktoren auf einer Weltkarte anzuzeigen, sowie die wichtigen Korrelationen auf Scatterplots zu veröffentlichen. Am Ende des Projektes haben wir uns dann noch entschlossen die unterstützten Länder anzuzeigen, um den Fortschritt dieser Länder zu visualisieren.

Durch die animierte Karte gemappt mit der Lebenserwartung jedes Landes können Trends über mehrere Jahre, sowie Länder die Unterstützung gebrauchen ausgelesen werden. Da die Karten interaktiv sind, kann man Zoomen, die Karte bewegen und wenn man über ein Land hovert, sieht man die einzelnen Statistiken des Landes noch genauer.

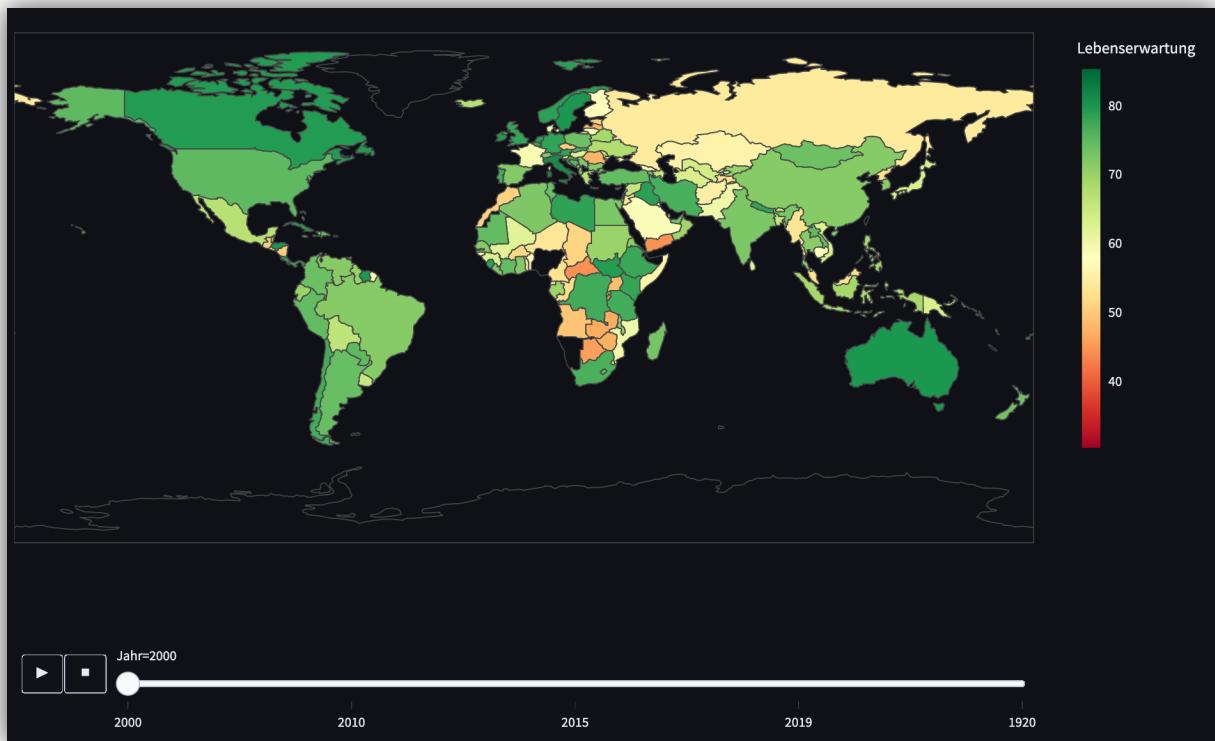


Abbildung 15: Lebenserwartung Karte (Quelle: Eigene Anfertigung)

Durch die Streudiagramme erkennt man gut den Zusammenhang zwischen den einzelnen Faktoren. Durch die Auswahl des Jahres, kann man hier auch gewisse Trends erkennen. Die Visualisierung helfen auch dabei zu erkennen, welches Land weit unter dem Durchschnitt der **einzelnen Kontinente** liegt. Somit kann man schnell erkennen, welches **Land noch am meisten Unterstützung braucht**.



Abbildung 16: Scatter Plot Lebenserwartung versus Trinkwasser Zugang (Quelle: Eigene Anfertigung)

Mit der Visualisierung der Länder, welche unterstützt wurden von den NPOs, sieht man, welche Auswirkungen die NPOs auf verschiedene Länder über einen Zeitraum hatten. Mit Hilfe dieser Visualisierung wird aufgezeigt, dass die den Ländern, welche Unterstützung bekommen haben, auch tatsächlich **Fortschritte** gemacht haben.



Abbildung 17: Scatterplot Lebenserwartung - Trinkwasser Zugang gefiltert der unterstützenden Organisationen (Quelle: Eigene Anfertigung)

Im Abschnitt 17.5 Visualisierungen sieht man noch weitere Visualisierungen, die auf der Webseite zu sehen sind. Diese haben wir jedoch nicht alle beschrieben, trotzdem stellen sie für NPOs einen Mehrwert für die Entscheidungsfindung dar.

## 11 Sicherheitsaspekte

Datensicherheit ist ein wichtiges Thema. In diesem Abschnitt erklären wir, wie die Sicherheit unserer Datenbank gewährleistet werden kann und welche Risiken trotz Schutzmassnahmen auftreten können.

### 11.1 Datenbank Server

Der Datenbank Server ist ausgezeichnet geschützt: Er befindet sich im Enterprise Lab der Hochschule Luzern, einem gut gesicherten Rechenzentrum. Es handelt sich um einen Windows Server 2022 (Datacenter Edition) mit allen installierten Patches und Updates. Die Firewall Regeln sind restriktiv, sodass auch nur die absolut nötigen Ports von aussen zugänglich sind. Auf dem Windows Server haben wir unseren MySQL Server installiert. Die Kommunikation erfolgt über eine sichere VPN-Verbindung. Somit ist diese verschlüsselt und Passwörter können nicht im Klartext abgehört werden.

### 11.2 Authentisierung und Autorisierung

Es muss grundsätzlich zwischen Datenschutz und Datensicherheit unterschieden werden.<sup>7</sup> Datenschutz bezeichnet den Schutz einer Person vor den Daten. Die Datensicherheit hingegen behandelt den Schutz der Daten vor Personen. Da wir in unserem Use Case keinerlei Kundendaten in der Datenbank speichern, müssen wir uns nicht um den Datenschutz kümmern. Die folgenden Umschreibungen beziehen sich daher immer auf die Datensicherheit.

Eine erste Massnahme zur Sicherstellung der Datensicherheit ist die Authentisierung und Autorisierung. Grundsätzlich ist unser Dashboard für alle Interessierten ausführbar, wenn sie das Skript besitzen. Es ist dabei nicht nötig und daher gar nicht möglich, sich auf der Plattform anzumelden. Jeder Benutzer und jede Benutzerin erhalten dieselben Zugriffsrechte. Zudem ist es so, dass die Endbenutzer keine Änderungen an der Datenbank vornehmen dürfen. Sie sollen die bestehenden Daten lediglich abfragen können. Das macht die Erstellung der Benutzer des Datenbanksystems sehr einfach.

Hinzufügend, haben wir für unser Datenbank ein starkes Passwort gewählt, welches wir mit dem «validate\_password» Plugin überprüft haben. Dadurch werden unautorisierte Zugriffe durch Bruteforcing enorm eingeschränkt, da es Billionen Jahre dauern würde das Passwort zu «cracken». Das Skript dazu ist im Anhang ersichtlich unter «Passwort Plugin Query».

---

<sup>7</sup> (Reisswolf, 2021)

### 11.3 Benutzer

Grundsätzlich haben wir uns für drei verschiedene Benutzer entschieden:  
**admin**

The screenshot shows the 'Users and Privileges' section of MySQL Workbench. On the left, a tree view shows 'User Accounts' with 'admin' selected. The main panel displays 'Details for account admin@%'. It shows the 'User' column with 'admin' and 'From Host' with '%'. Below this is a table of grants:

User	From Host
dashboard	%
developer	%
mysql.infoschema	localhost
mysql.session	localhost
mysql.sys	localhost
read_user	%
read_user	localhost
root	localhost

On the right, there are tabs for 'Login', 'Account Limits', 'Administrative Roles', and 'Schema Privileges'. Under 'Administrative Roles', several checkboxes are checked, including DBA, MaintenanceAdmin, ProcessAdmin, UserAdmin, SecurityAdmin, MonitorAdmin, DBDesigner, DBManager, ReplicationAdmin, and BackupAdmin. Under 'Schema Privileges', a large list of privileges is shown, mostly checked, such as ALTER, CREATE, INDEX, INSERT, SELECT, SHOW DATABASES, and SUPER.

Abbildung 18: Benutzer: admin (Quelle Eigene Anfertigung)

“admin” ist ein Benutzer, welcher alle Rechte auf dem Datenbanksystem erhält. Der “admin” Benutzer ist dementsprechend besonders schützenswert, da mit ihm am meisten Schaden angerichtet werden kann.

Der “admin” Benutzer wird lediglich dann verwendet, wenn administrative Änderungen am Datenbanksystem getätigt werden müssen. Jedoch ist es das Ziel, diesen Benutzer nicht für andere Arbeiten zu verwenden, um unnötiges Risiko zu vermeiden.

### developer

The screenshot shows the 'Users and Privileges' section of MySQL Workbench. On the left, a tree view shows 'User Accounts' with 'developer' selected. The main panel displays 'Details for account developer@%'. It shows the 'User' column with 'developer' and 'From Host' with '%'. Below this is a table of grants:

User	From Host
dashboard	%
developer	%
mysql.infoschema	localhost
mysql.session	localhost
mysql.sys	localhost
read_user	%
read_user	localhost
root	localhost

On the right, there are tabs for 'Login', 'Account Limits', 'Administrative Roles', and 'Schema Privileges'. Under 'Schema Privileges', a table shows 'Schema' as 'dankdaten' and 'Privileges' as 'ALTER, ALTER ROUTINE, CREATE, CREATE ROUTINE, CREATE TEMPORARY TABLES, CREATE VIEW, DELETE, DROP, EVENT, EXECUTE, FILE, GRANT OPTION, INDEX, INSERT, LOCK TABLES, PROCESS, REFERENCES, RELOAD, REPLICATION CLIENT, REPLICATION SLAVE, SELECT, SHOW DATABASES, SHOW VIEW, SHUTDOWN, SUPER'. Under 'Object Rights' and 'Other Rights', several checkboxes are listed, mostly unchecked.

Abbildung 19: Benutzer: developer (Quelle Eigene Anfertigung)

Für die gesamte Entwicklung unseres Produktes haben wir den Benutzer “developer” angelegt. Dieser Benutzer hat genügend Berechtigungen, um in der Entwicklung nicht eingeschränkt zu sein. Grundsätzlich hat der Benutzer Berechtigungen, um mit der Datenbank “dankdaten” arbeiten zu können. Anders als der “admin” Benutzer ist es dem “developer” Benutzer jedoch nicht möglich, schwerwiegende administrative Anpassungen am Datenbanksystem zu tätigen.

## dashboard

The screenshot shows the MySQL 'Users and Privileges' interface. On the left, a list of users is displayed with 'dashboard' selected. The main panel shows 'Details for account dashboard@%'. It includes tabs for 'Login', 'Account Limits', 'Administrative Roles', and 'Schema Privileges'. Under 'Schema Privileges', it lists 'dankdaten' with 'SELECT' privilege. Below this, there are sections for 'Object Rights' (with checkboxes for SELECT, INSERT, UPDATE, DELETE, EXECUTE, and SHOW VIEW), 'DDL Rights' (with checkboxes for CREATE, ALTER, CREATE INDEX, INDEX, CREATE VIEW, CREATE ROUTINE, ALTER ROUTINE, EVENT, TRIGGER, and DROP), and 'Other Rights' (with checkboxes for GRANT OPTION, CREATE TEMPORARY TABLES, and LOCK TABLES). Buttons at the bottom include 'Revoke All Privileges', 'Delete Entry', and 'Add Entry'.

Abbildung 20: Benutzer: dashboard (Quelle Eigene Anfertigung)

Schlussendlich haben wir uns noch für einen dritten Benutzer entschieden. Somit können die NPO's oder die Interessierenden, auf unserer Webplattform Abfragen tätigen. Durch die starke Einschränkung des Webbenutzers können wir auch gleich die Gesamtsicherheit des Datenbanksystems verbessern. Auch falls eine SQL-Injektion möglich wäre, kann kein grosser Schaden angerichtet werden, da die Datenbank alle Anfragen, die über ein "SELECT" hinausgehen, einfach zurückweist.

### 11.4 Zusätzliche Sicherheit

Wir verwenden MySQL 8<sup>8</sup>, dies hat nicht nur den Vorteil, dass die neusten Bug- und Exploitfixes verwendet werden, sondern auch die Standardisierung des 'caching\_sha2\_password'-Plugins, welches zur Authentisierung ein SHA-2 Algorithmus mit 256-bit Passwortverschlüsselung verwendet.<sup>9</sup>

Für die sichere Verbindung zu unserer Datenbank wird das SSL-Protokoll TLS\_AES\_256\_GCM\_SHA384 verwendet, dies verschlüsselt die Datenübertragung zwischen Client und Server.

Mithilfe von Backups werden die Struktur und die Daten unserer Datenbank gesichert und durch einen Cronjob können wir dies sogar automatisieren, um tägliche Backups zu erstellen, nämlich ein Backup ist das absolut wichtigste unserer Meinung nach. Falls etwas schief gehen sollte, kann man immer auf einen früheren Punkt zugreifen und es besteht nicht die Gefahr, dass die ganze Datenbank zunichte geht und man von Null anfangen muss.

<sup>8</sup> (MySQL, 2022)

<sup>9</sup> (MySQL, 2022)

## 11.5 Risiken

Die Angriffsflächen wurden minimiert, indem möglichst viele Faktoren, welche die Sicherheit beeinflussen können, schon beseitigt worden sind. Es kann trotzdem sein, dass Angreifer Zugriff auf das System bekommen könnten, indem sie den Computer eines Administrators mit Malware infizieren und sich somit Zugriff verschaffen. Durch die 2-Faktor-Authentifizierung seitens der Hochschule können wir auch davon ausgehen, dass Externe Leute nicht auf das System zugreifen können.

Sicherheitsmassnahmen sind stündliche Snapshots und tägliche Backups der Datenbank. Auch wenn ein Angreifer Daten löschen oder manipulieren kann, befinden sich immer sehr aktuelle Backups auf einem anderen Datenträger. Das stellt ebenfalls die Integrität der Daten sicher.

## 11.6 Auditing

Wir haben uns in diesem Projekt bewusst gegen ein Audit Log entschieden aus folgenden Gründen: Ein Audit Log macht meistens dann Sinn, ...

1. ... wenn Daten geändert werden. In unserem Projekt werden jedoch keine Datenbanktransaktionen vorgenommen.
2. ... wenn mehrere Benutzer existieren, um nachzuvollziehen wer was angeschaut hat. In unserem Projekt gibt es jedoch nur einen Benutzer welcher Abfragen macht, daher wäre ein Log äusserst uninteressant.
3. ... wenn vertraulicher Daten vorhanden sind. In unserer Datenbank befinden sich jedoch nur öffentlich Zugängliche Daten, welche von uns aufbereitet wurden.

## 12 Tests

Um unsere Applikation vollumfänglich zu testen, haben wir nachfolgend noch diverse Bereiche unserer Implementation genauer untersucht. Unser Testkonzept gliedert sich in funktionale Tests, welche sich mit der Systemfunktionalität auseinandersetzen, in Performancetests, um allfällige Verbesserungen bezüglich der Laufzeit zu erkennen, sowie in Sicherheitstest, welche mögliche Gefahren erkennen, und passende Massnahmen definieren.

### 12.1 Funktionale Tests

Testfall	Ergebnis
“Dropdown” enthalten vorgesehene Werte	✓
Kartendiagramm wird anhand verknüpften Query korrekt dargestellt	✓
Streudiagramm wird anhand verknüpften Query korrekt dargestellt	✓
Diagramme sind skalierbar	✓
Legende sind gut ersichtlich und nachvollziehbar	✓
Farbkodierung ist gut gewählt und immer gleichbleibend zu den Werten	✓
Die einzelnen Diagramme sind interaktiv: Man kann auf die Karten- sowie Streudiagramme schweben und es zeigt die wichtigsten Daten an.	✓

Aus technischer Sicht sind wir mit unserem Endprodukt sehr zufrieden. Wir haben keine Funktionalitäten gefunden die Fehler aufweisen, oder mangelhaft implementiert wurden. Auch finden wir, dass die Interaktion für den User gut verständlich ist.

### 12.2 Performancetest

Wir haben bei der Erstellung unserer Querys auf gewisse Dinge beachtet, welche eine Datenbank verlangsamen könnten. Einerseits haben wir versucht Ausdrücke zu vermeiden, welche einen ‘Full-Scan’ in einer Tabelle durchführen. Wir haben ebenfalls den Ausdruck ‘SELECT \*’ vermieden, da dadurch unnötige Kolumnen geladen werden können, welche die Performance beeinträchtigen. Beim Zusammenführen von Tabellen haben wir darauf geschaut, dass wir keine «OUTER JOIN»-Statements verwenden, da diese ebenfalls die Query-Exekution verlangsamen.

Um die Performance zu erhöhen haben wir oft mit Primary-Keys gearbeitet, da aber unsere Datenbank nicht gross ist, haben wir keine Unterschiede gemerkt.

Zur Erinnerung kurz noch einmal erwähnt: Unsere Datenbank umfasst “nur” ca. 3'000 Datensätze, sowie 7 Tabellen. Dies ist aus Sicht der Datenbank keine riesige Menge, weswegen die Abfragen ohnehin schon schnell sind. Auch verfügen all unsere Tabellen über einen “Primary Key”, wodurch die Query Abfragen profitieren können, und bereits maximiert sind. Selbst verschachtelte “JOIN” Funktionen oder Abfragen mit einer “WHERE” Bedingung (z. B. “LIKE”), scheinen keinen spürbaren Unterschied zu machen.

### 12.3 Sicherheitstests

Durch die Massnahmen, welche bereits im Kapitel “Sicherheitsaspekte” beschrieben wurden, haben wir einige Massnahmen getroffen. Da sich in unserer Gruppe keine Sicherheitsexperten: innen befinden, können wir jedoch nicht komplett ausschliessen, dass ein Angreifer Schäden am System verursachen könnte.

## 13 Ergebnis

Mit der Webapplikation haben wir ein Analyse Tool für NPOs erschaffen. Durch verschiedene Visualisierungen mit denen gearbeitet werden, kann ein NPO einen Mehrwert daraus ziehen, dass sie nun schneller und einfacher Entscheidungen treffen können, welches Land sie in Zukunft unterstützen wollen und sollen. Zudem können sie mit den Visualisierungen den Sponsoren die Trends besser zeigen und zu überzeugen, dass gewisse Länder Unterstützung brauchen.

Durch die Visualisierung auf der Weltkarte, sieht man auf der Webapplikation direkt den Gesamtüberblick der ganzen Welt, wie es mit der Lebenserwartung, dem Trinkwasser Zugang und der Säuglingssterblichkeitsrate aussieht.

Mit Hilfe der Streudiagramme, bei denen die Lebenserwartung mit der Säuglingssterblichkeitsrate und dem Trinkwasserzugang verglichen wird, sieht man, dass es eine Korrelation zwischen den verschiedenen Faktoren gibt. Dies verhilft die Entscheidungsfindung der NPOs. Da die verschiedenen Visualisierungen durch die verschiedenen Jahre animiert sind, kann man auch gewisse Trends erkennen. Eine Hilfe bietet auch die Visualisierung von den Ländern, welche von den Top 5 NPOs unterstützt werden. Dadurch kann gesehen werden, wie sich die verschiedenen Länder entwickelt haben.

Zum Schluss wollen wir den NPOs für ihre Unterstützung, Engagement und den Beitrag die sie leisten, herzlich danken!

## 14 Reflexion

Wir konnten in dieser Arbeit einiges lernen und Wissenswertes mitnehmen. Wir haben uns im Nachhinein auch Gedanken gemacht, was nicht ganz rund lief und was wir besser hätte machen können. Wir haben uns aufgrund der Daten schnell für eine relationale Datenbank entschieden, hier hätten wir uns bereits am Anfang mehr Gedanken zu anderen Datenbank-Technologien wie eine Graph- oder Dokumentdatenbank machen können. Wir haben uns diese Gedanken zwar gemacht, jedoch zu einem Zeitpunkt, wo unsere Datenbank bereits lange bestand und die Auswertungen daraus resultierten, hier machte es zeitlich bedingt keinen Sinn alles nochmals neu aufzubauen. Ein weiteres Learning, welches bei einer Weiterführung des Projektes hilfreich wäre: Wir bräuchten mehr Daten von der WHO oder NPO, um bessere Aussagen machen zu können. Hierzu wäre eine Zusammenarbeit mit den NPO's von Vorteil. Nämlich online stehen begrenzte Daten zur Verfügung.

Positive Punkte, die wir hervorheben möchten: Besonders gelungen ist uns die Struktur des Projektes und dessen Ablauf. So haben wir uns zu Beginn ein Grundgerüst einer Dokumentation aufgebaut, um jederzeit zu sehen, welche Themen wir bereits erledigt haben und welche noch folgen, dies auch mit Hilfe des Semesterplans. Diese Dokumentation haben wir jede Woche mit den aktuellen Themen nachgeführt, was uns eine grosse Arbeitslast zum Ende des Semesters erspart hat. Zudem sind wir sehr zufrieden mit der Datenaufbereitung. Wir haben uns lange damit aufgehalten die richtigen Datenquellen zu suchen, damit auch alle Teammitglieder zufrieden sind. Beispielsweise, bei den Säuglingssterblichkeit Daten haben wir mehrere andere Datenquellen evaluiert. Wir hatten sehr viele Ideen und auch die nötigen Daten dazu, aber der Umfang des Projekts steigerte sich dementsprechend. Nachfolgend ein Auszug aus unserem Ideen Evaluation:

Idee	Daten
Bevölkerung, die sicher bewirtschaftete Abwasserentsorgungsdienste nutzt	<a href="#">Link</a>
Ärzte (pro 10.000 Einwohner)	<a href="#">Link</a>
Bevölkerung, die zumindest eine sanitäre Grundversorgung in Anspruch nimmt	<a href="#">Link</a>

Im Nachhinein konnten wir zum Glück feststellen, dass wir uns korrekt entschieden haben. Dank dieser zusätzlich investierten Zeit zu Beginn, konnten wir ohne jegliche Probleme mit der Datenstruktur das Projekt durchführen.

Den Aufbau der Datenbank haben wir in einzelnen SQL-Skripts festgehalten, dies hat zur Folge, dass es relativ einfach war, die Verfügbarkeit der Daten sicher zu stellen. Während der Entwicklungsphase konnten wir bei Verlusten die Datenbank anhand unserer Skripts problemlos wiederherstellen. Das Arbeiten mit Skripts ermöglichte uns, eine Test-Instanz zu installieren, an welcher einzelne Arbeiten ausgefeilt werden konnten, bevor sie auf der Haupt-Instanz eingespielt wurden.

### Zeitplan

Obwohl wir anfangs viel Zeit für die Eruierung des Use Case aufgewendet haben, sind wir in den ersten Wochen zügig mit dem Projekt vorangekommen. Das Datenbankmodell anschliessend anhand des Use Case Entscheids entworfen sowie in die Datenbank importiert. Die Datenbereinigung haben wir in der Gruppe zusammen rasch bearbeitet, um dann das Python Skript für den Datenimport vorbereiten zu können. Glücklicherweise hatten wir auch da keine längerfristigen Probleme, weswegen wir gleich die ersten Queries testen konnten. Schon bald ging es an die ersten "JOIN" Überlegungen und Anwendungen, wie wir den Output genau

ausgeben beziehungsweise visualisieren möchten. Zusammengefasst lässt sich sagen, dass wir zu Beginn sehr effizient am Projekt gearbeitet haben. Dementsprechend blieben zum Ende nur noch die Fertigstellung der schriftlichen Dokumentation übrig.

#### 14.1 Choekyel N.

Mir persönlich gefiel es, dass wir unsere Meetings jeden Sonntag online abhielten und damit effizient und zielbringend zusammenarbeiten konnten. Die offene und hilfsbereite Teamarbeit der einzelnen Mitglieder hat definitiv dazu beigetragen, Spass sowie auch Interesse während des ganzen Moduls aufrechtzuerhalten.

#### Learnings aus dem Modul DBS

In diesem Modul konnte ich das erste Mal mit einer Datenbank tief auseinandersetzen. Mir wurde unter anderem die Wichtigkeit der Diversität in einem Team bewusst. Wir hatten das Glück, dass wir einen Spezialisten oder eine Spezialistin für jeden Bereich der Arbeit aufweisen konnten. Dieser Umstand hat die Projektarbeit sehr vorangetrieben. Weiter haben wir seit Beginn der Arbeit einen strikten Zeitplan geführt. Dadurch fanden stets produktive Treffen statt, um am Projekt zu arbeiten. Infolgedessen konnten wir unsere Ziele einhalten und das Projekt erfolgreich abschliessen.

#### 14.2 Oliver G.

Ich empfand die Teamarbeit als sehr angenehm. Wir hatten immer wieder spontane und spannende Inputs während unserer Teammeetings, welche stets für Staunen sorgten. Insbesondere gefielen mir auch unsere wöchentlichen Besprechungen am Sonntagnachmittag. Dadurch konnten wir konstant immer wieder gemeinsam an unserem Projekt feilen und die Idee weiter vorantreiben. Trotz des guten Zeitplans kam gegen den Schluss eine intensive Phase.

#### Learnings aus dem Modul DBS

Da ich noch keine Erfahrung mit NoSQL und Graph Datenbanken hatte, konnte ich von diesen Themenbereichen sehr viel lernen. Durch das Projekt konnte ich mein grundlegendes Wissen, das ich über SQL-Datenbanken hatte, nochmals vertiefen. Durch die Teamarbeit ist mir klar geworden, wie wichtig ein geordneter Zeitplan im Projekt ist. Auch ist mir aufgefallen, dass regelmässige Treffen mit den Projektkollegen ein Muss ist, um ein Projekt erfolgreich abzuschliessen.

#### 14.3 Tharrmeehan K.

Ich fand, dass unsere Teamarbeit gut verlief. Sehr gewinnbringend fand ich, dass wir immer den aktuellen Stand der anderen Teammitglieder kannten. Die wöchentlichen Meetings fand ich super, da wir uns austauschten und unsere Aufgaben neu aufteilen konnten. Dies steigerte die Effizienz stark. Das Projekt war aus meiner Sicht ein Erfolg, da wir unsere Ideen trotz des kurzen Zeitraums umsetzen konnten. Unser Git-Repository fand ich auch super, da wir alles wichtige Daten, sowie Dokumente zentral hatten und wichtige Änderungen direkt comitten konnten.

#### Learnings aus dem Modul DBS

Als Anfänger im Bereich der Datenbanktechnik konnte ich sehr vieles aus den einzelnen Semesterwochen mitnehmen, für meinen späteren Werdegang wird es wichtig sein, dass ich diese Kenntnisse habe. Die wöchentlichen Meetings waren sehr wichtig meiner Meinung zur Aufgabenteilung.

## 15 Literaturverzeichnis

- Bansal, S., & Kagemann, S. (März 2015). Integrating Big Data: A Semantic Extract-Transform-Load Framework. *Computer*, S. 42-50.
- Jalli, A. (kein Datum). Why Is Python So Popular? *codingem.com*, S. 1.
- Luber, S., & Litzel, N. (29. Oktober 2020). Was ist Streamlit? *BIGDATA INSIDER*, S. 1.
- Reisswolf. (16. Juni 2021). *Datenschutz oder Datensicherheit? Wir klären auf.* Von REISSWOLF: <https://www.reisswolf.ch/news/datenschutz-oder-datensicherheit-wir-klaeren-auf/> abgerufen
- Rudnicka, J. (28. 11 2022). *Statista.com*. Von <https://de.statista.com/themen/47/lebenserwartung/#topicOverview> abgerufen
- Rudnicka, J. (28. November 2022). *Statistiken zum Thema Lebenserwartung*. Von Statista: [https://de.statista.com/themen/47/lebenserwartung/#topicHeader\\_wrapper](https://de.statista.com/themen/47/lebenserwartung/#topicHeader_wrapper) abgerufen

## 16 Abbildungsverzeichnis

Abbildung 1: Lebenserwartung von [1920, 2000, 2010, 2015, 2019] (Quelle: Eigene Anfertigung) .....	5
Abbildung 2: Trinkwasserzugang von [2000-2017] (Quelle: Eigene Anfertigung) .....	6
Abbildung 3: Säuglingssterblichkeit von [1950-2019] (Quelle: Eigene Anfertigung)....	6
Abbildung 4: 5 Non-Profit-Organisationen (Quelle Eigene Anfertigung).....	7
Abbildung 5: Modell Systemarchitektur (Quelle: Eigene Anfertigung) .....	8
Abbildung 6: Entity-Relationship-Diagramm (Quelle: Eigene Anfertigung).....	9
Abbildung 7: Entity-Relationship-Modell der Datenbank (Quelle: Eigene Anfertigung) .....	10
Abbildung 8: Auszug vom CREATE TABLE (Quelle: Eigene Anfertigung).....	11
Abbildung 9: Prozess von den Rohdaten bis in unsere Datenbank (Quelle: Eigene Anfertigung) .....	12
Abbildung 10: Select Switzerland ohne Indexierung (Quelle: Eigene Anfertigung) ...	13
Abbildung 11: Select Switzerland mit Indexierung (Quelle: Eigene Anfertigung) .....	13
Abbildung 12: Architekturübersicht unserer Datenvisualisierung (Quelle: Eigene Anfertigung) .....	15
Abbildung 13: View für die Daten der Schweiz (Quelle: Eigene Anfertigung) .....	16
Abbildung 14: View für die Statistiken jedes Landes (Quelle: Eigene Anfertigung)...	16
Abbildung 15: Lebenserwartung Karte (Quelle: Eigene Anfertigung) .....	17
Abbildung 16: Scatter Plot Lebenserwartung versus Trinkwasser Zugang (Quelle: Eigene Anfertigung) .....	18
Abbildung 17: Scatterplot Lebenserwartung - Trinkwasser Zugang gefiltert der unterstützenden Organisationen (Quelle: Eigene Anfertigung) .....	18
Abbildung 18: Benutzer: admin (Quelle Eigene Anfertigung) .....	20
Abbildung 19: Benutzer: developer (Quelle Eigene Anfertigung) .....	20
Abbildung 20: Benutzer: dashboard (Quelle Eigene Anfertigung) .....	21

## 17 Anhang

### 17.1 CREATE & LOAD Queries

```

1 -----
2 -- create continent table
3 -----
4 CREATE TABLE continent
5 (
6 continent_ID INT NOT NULL AUTO_INCREMENT,
7 continent varchar(55),
8 PRIMARY KEY (continent_ID)
9 );
10 -----
11 -- load continent csv in table
12 -----
13 LOAD DATA LOCAL INFILE '/Users/oliver/HSLU/DBS/csv/transform/continents.csv'
14 INTO TABLE continent FIELDS ENCLOSED BY '\"' ESCAPED BY '\\\' LINES TERMINATED
15 BY '\n' IGNORE 1 LINES (continent);
16 -----
17 -- create country table
18 -----
19 CREATE TABLE country
20 (
21 country_ID INT NOT NULL AUTO_INCREMENT,
22 country VARCHAR(35),
23 population_estimate INT,
24 gdp_estimate INT,
25 iso_a3 VARCHAR(3),
26 continent_ID INT,
27 PRIMARY KEY (country_ID),
28 FOREIGN KEY (continent_ID) REFERENCES continent(continent_ID)
29 );
30 -----
31 -- load country csv in table
32 -----
33 LOAD DATA LOCAL INFILE '/Users/oliver/HSLU/DBS/csv/transform/countries.csv'
34 INTO TABLE country FIELDS TERMINATED BY ',' OPTIONALLY ENCLOSED BY '\"' ESCAPED
35 BY '\\\' LINES TERMINATED BY '\n' IGNORE 1 LINES (country, population_estimate,
36 gdp_estimate, iso_a3, continent_ID);
37 -----
38 -- create life_expectancy table
39 -----
40 CREATE TABLE life_expectancy
41 (
42 country_ID INT,
43 period YEAR,
44 percentage_life DECIMAL(5,2),
45 PRIMARY KEY (country_ID, period),
46 FOREIGN KEY (country_ID) REFERENCES country(country_ID));
47 -----
48 -- load life expectancy csv in table
49 -----
50 LOAD DATA LOCAL INFILE '/Users/oliver/HSLU/DBS/csv/transform/life_expectancy.csv'
51 INTO TABLE life_expectancy FIELDS TERMINATED BY ',' OPTIONALLY ENCLOSED
52 BY '\"' ESCAPED BY '\\\' LINES TERMINATED BY '\n'
53 IGNORE 1 LINES (country_ID, period, percentage_life);
54 -----
55 -- create drinking_water_supply table
56 -----
57 CREATE TABLE drinking_water_supply
58 (
59 country_ID INT,
60 period YEAR,
61 percentage_water DECIMAL(5,2),
62 PRIMARY KEY (country_ID, period),
63 FOREIGN KEY (country_ID) REFERENCES country(country_ID));

```

```

64 -----  

65 -- load drinking water supply csv in table  

66 -----  

67 LOAD DATA LOCAL INFILE '/Users/oliver/HSLU/DBS/csv/transform/drinking_water_supply.csv'  

68 INTO TABLE drinking_water_supply FIELDS TERMINATED BY ',' OPTIONALLY ENCLOSED  

69 BY '\"' ESCAPED BY '\\\' LINES TERMINATED BY '\n'  

70 IGNORE 1 LINES (country_ID, period, percentage_water);  

71 -----  

72 -- create infant_mortality_rate table  

73 -----  

74 CREATE TABLE infant_mortality_rate  

75 (  

76 country_ID INT,  

77 period YEAR,  

78 percentage_infant DECIMAL(5,2),  

79 PRIMARY KEY (country_ID, period),  

80 FOREIGN KEY (country_ID) REFERENCES country(country_ID));  

81 -----  

82 -- load infant mortality rate csv in table  

83 -----  

84 LOAD DATA LOCAL INFILE '/Users/oliver/HSLU/DBS/csv/transform/infant_death_rate.csv'  

85 INTO TABLE infant_mortality_rate FIELDS TERMINATED BY ',' OPTIONALLY ENCLOSED  

86 BY '\"' ESCAPED BY '\\\' LINES TERMINATED BY '\n'  

87 IGNORE 1 LINES (country_ID, period, percentage_infant);  

88 -----  

89 -- create organisation table  

90 -----  

91 CREATE TABLE organisation(  

92 organisation_id INT NOT NULL AUTO_INCREMENT,  

93 organisation VARCHAR(55) NOT NULL,  

94 PRIMARY KEY (organisation_id)  

95 );  

96 -----  

97 -- insert top 5 organisations in table  

98 -----  

99 INSERT INTO organisation(organisation)  

100 VALUES ('Blood:Water'), ('Pure Water for the World'),  

101 ('water for good'), ('splash'), ('Lifewater');  

102 -----  

103 -- create country_organisation table  

104 -----  

105 CREATE TABLE country_organisation(  

106 organisation_id INT NOT NULL,  

107 country_id INT NOT NULL,  

108 PRIMARY KEY (organisation_id, country_id),  

109 FOREIGN KEY (organisation_id) REFERENCES organisation(organisation_id),  

110 FOREIGN KEY (country_id) REFERENCES country(country_id)  

111 );  

112 -----  

113 -- insert country id from organisation in table  

114 -----  

115 INSERT INTO country_organisation(organisation_id, country_id)  

116 VALUES (1, 60), (1, 90), (1, 103), (1, 181);  

117 -----  

118 INSERT INTO country_organisation(organisation_id, country_id)  

119 VALUES (2, 75), (2, 76);  

120 -----  

121 INSERT INTO country_organisation(organisation_id, country_id)  

122 VALUES (3, 33);  

123 -----  

124 INSERT INTO country_organisation(organisation_id, country_id)  

125 VALUES (4, 14), (4, 30), (4, 36), (4, 60), (4, 79), (4, 121),  

126 (4, 172), (4, 191);  

127 -----  

128 INSERT INTO country_organisation(organisation_id, country_id)  

129 VALUES (5, 30), (5, 60), (5, 181), (5, 185);  

130 -----  

131

```

## 17.2 VIEW Queries

```

● ● ●

1 CREATE VIEW water
2 AS SELECT cn.country, period, percentage_water, iso_a3
3 FROM drinking_water_supply AS dw
4 INNER JOIN country AS cn ON dw.country_ID = cn.country_id;
5
6 CREATE VIEW life
7 AS SELECT cn.country, period, percentage_life, iso_a3
8 FROM life_expectancy AS le
9 INNER JOIN country AS cn ON le.country_ID = cn.country_id;
10
11 CREATE VIEW infant
12 AS SELECT cn.country, period, percentage_infant, iso_a3
13 FROM infant_mortality_rate AS im
14 INNER JOIN country AS cn ON im.country_ID = cn.country_id;
15
16 CREATE VIEW water_life_infant
17 AS SELECT cn.population_estimate, continent.continent, cn.iso_a3, cn.gdp_estimate,
18 cn.country, dw.period, dw.percentage_water, le.percentage_life, im.percentage_infant
19 FROM country AS cn
20 LEFT JOIN continent ON cn.continent_ID = continent.continent_ID
21 LEFT JOIN drinking_water_supply AS dw ON cn.country_ID = dw.country_ID
22 LEFT JOIN infant_mortality_rate AS im ON dw.country_ID = im.country_ID
23 AND dw.period = im.period
24 INNER JOIN life_expectancy AS le ON im.country_ID = le.country_ID
25 AND im.period = le.period;
26
27 CREATE VIEW water_life
28 AS SELECT cn.population_estimate, continent.continent, cn.iso_a3,
29 cn.gdp_estimate, cn.country, dw.period, dw.percentage_water, le.percentage_life
30 FROM country AS cn
31 LEFT JOIN continent ON cn.continent_ID = continent.continent_ID
32 LEFT JOIN drinking_water_supply AS dw ON cn.country_ID = dw.country_ID
33 INNER JOIN life_expectancy AS le ON dw.country_ID = le.country_ID
34 AND dw.period = le.period;
35
36 CREATE VIEW water_infant
37 AS SELECT cn.population_estimate, continent.continent, cn.iso_a3,
38 cn.gdp_estimate, cn.country, dw.period, dw.percentage_water, im.percentage_infant
39 FROM country AS cn
40 LEFT JOIN continent ON cn.continent_ID = continent.continent_ID
41 LEFT JOIN drinking_water_supply AS dw ON cn.country_ID = dw.country_ID
42 LEFT JOIN infant_mortality_rate AS im ON dw.country_ID = im.country_ID
43 AND dw.period = im.period;
44
45 CREATE VIEW choose_switzerland
46 AS SELECT wln.population_estimate, wln.continent, wln.iso_a3, wln.gdp_estimate,
47 wln.country, wln.period, wln.percentage_water, wln.percentage_life, wln.percentage_infant
48 FROM water_life_infant AS wln
49 WHERE wln.country = 'Switzerland';
50
51 CREATE VIEW organisation_country
52 AS SELECT cn.population_estimate, continent.continent, cn.iso_a3, cn.gdp_estimate,
53 cn.country, og.organisation, wl.percentage_water, wl.percentage_life, wl.period
54 FROM country_organisation AS co
55 LEFT JOIN country AS cn ON co.country_id = cn.country_id
56 LEFT JOIN organisation AS og ON co.organisation_id = og.organisation_id
57 LEFT JOIN continent ON cn.continent_id = continent.continent_id
58 INNER JOIN water_life AS wl ON cn.iso_a3 = wl.iso_a3;

```

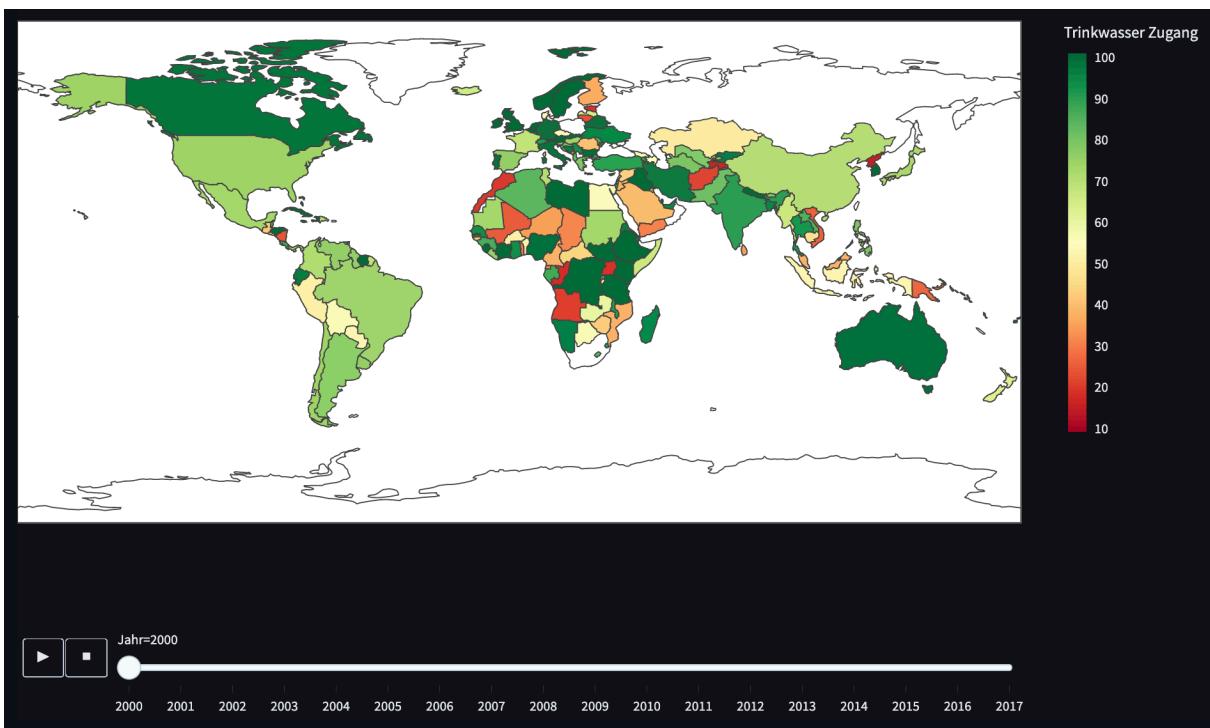
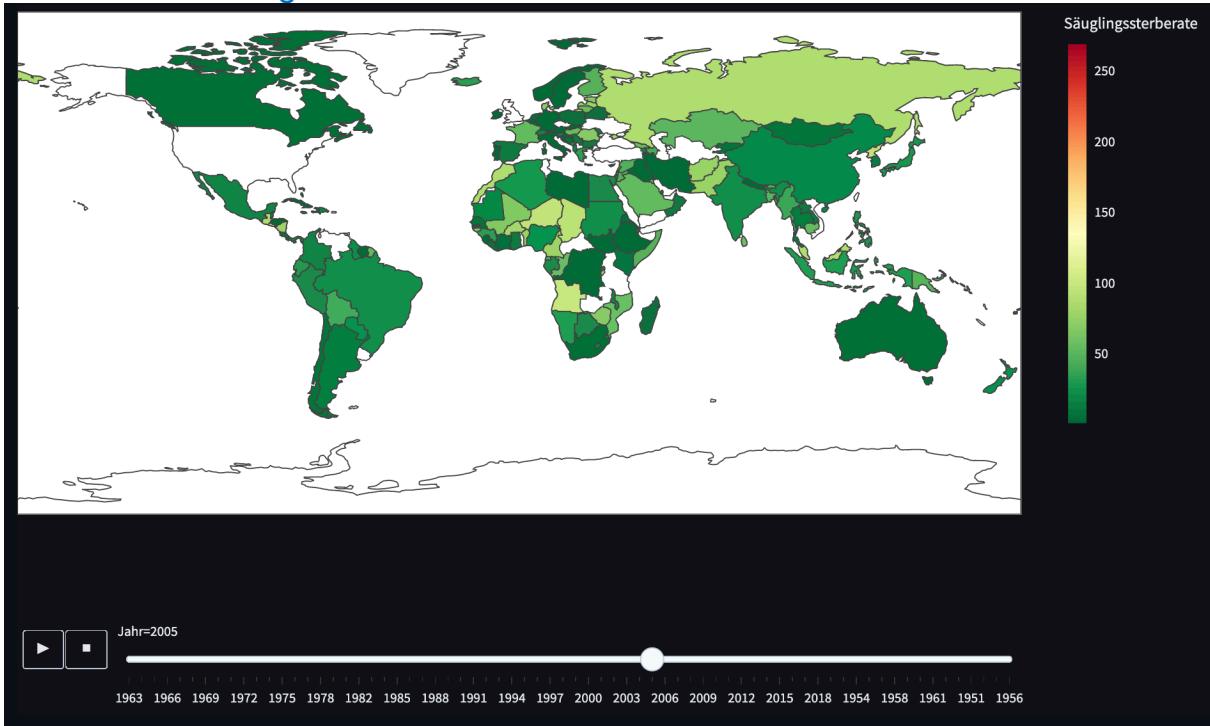
## 17.3 Passwort Plugin Query

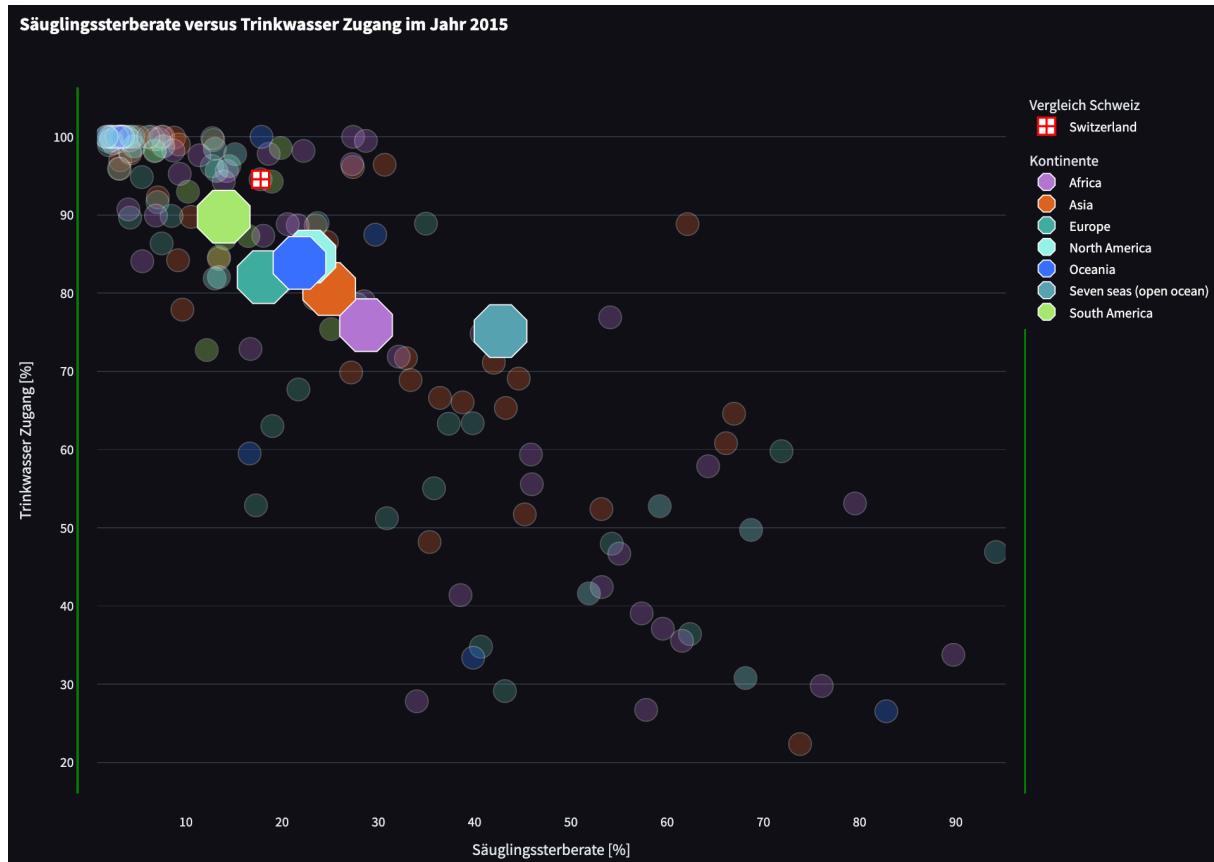
```
● ● ●  
1 INSTALL PLUGIN validate_password SONAME 'validate_password';  
2  
3 SELECT PLUGIN_NAME, PLUGIN_STATUS FROM INFORMATION_SCHEMA.PLUGINS WHERE PLUGIN_NAME LIKE 'validate%';  
4  
5 SELECT VALIDATE_PASSWORD_STRENGTH('Crj@BbgKvkrQv&N&J45e')
```

## 17.4 USER Queries

```
● ● ●  
1 CREATE USER developer IDENTIFIED BY 'q~vUYe@-gJ>5e2;{o}x!';  
2 GRANT ALL ON dankdaten.* TO developer;  
3 FLUSH PRIVILEGES;  
4  
5  
6 CREATE USER dashboard IDENTIFIED BY 'p~bUYk@-fJ>4e7;{s}x@';  
7  
8 GRANT SELECT ON dankdaten.* TO dashboard;  
9  
10 GRANT SELECT  
11     ON dankdaten.water_life_infant  
12     TO dashboard;  
13 GRANT SELECT  
14     ON dankdaten.water_life  
15     TO dashboard;  
16 GRANT SELECT  
17     ON dankdaten.water_infant  
18     TO dashboard;  
19 GRANT SELECT  
20     ON dankdaten.choose_switzerland  
21     TO dashboard;  
22 GRANT SELECT  
23     ON dankdaten.organisation_country  
24     TO dashboard;  
25 GRANT SELECT  
26     ON dankdaten.life  
27     TO dashboard;  
28 GRANT SELECT  
29     ON dankdaten.water  
30     TO dashboard;  
31 GRANT SELECT  
32     ON dankdaten.infant  
33     TO dashboard;  
34  
35 FLUSH PRIVILEGES;
```

## 17.5 Visualisierungen





## 17.6 INDEX Query



```
1 ALTER TABLE country ADD UNIQUE INDEX index_country( country );
```

## 18 Projektplan

Projektplan				
Thema	Details	Status	Stichtag	Resultat
Follow-Up Gruppe; Zoom	<a href="https://hslu.zoom.us/j/7349185514">https://hslu.zoom.us/j/7349185514</a>	Erfedigt	jeden Sonntag	
<b>Datenbanksysteme Woche 1</b>				
Gruppeneinteilung Projekte		Erfedigt		Oliver Grun, Choekyel Nyungmartsang, Tharmeehan Krishnathasan
Kurze Reflektion durch alle Teammitglieder	Lernrückblick dokumentieren	Erfedigt		
Miro Board erstellen	<a href="https://miro.com/login/">https://miro.com/login/</a>	Erfedigt		
Leistungsachweis anschauen	ILIAS	Erfedigt		
<b>Datenbanksysteme Woche 2</b>				
Brainstorming welchen Datenbank	Lernrückblick dokumentieren	Erfedigt		MySQL
Brainstorming Daten	<a href="https://www.kaggle.com/datasets/utkarshxy/who-world-health-statistics-2020-complete">https://www.kaggle.com/datasets/utkarshxy/who-world-health-statistics-2020-complete</a>	Erfedigt		
Verbindung VPN	HSLU / ILIAS	Erfedigt		Pulse Secure
MySQL Server und Workbench installieren	Anleitung in SW01	Erfedigt		
<b>Datenbanksysteme Woche 3</b>				
Brainstorming Themenwahl		Erfedigt		
Brainstorming Probleme für Projekt		Erfedigt		
Brainstorming Challenges		Erfedigt		
Brainstorming Interessen		Erfedigt		
Kurze Reflektion durch alle Teammitglieder	Lernrückblick dokumentieren	Erfedigt		
<b>Datenbanksysteme Woche 4</b>				
Coaching	Gespräch mit Herr Kaufmann (offene Fragen klären)	Erfedigt		Feedback: Gute Idee
Entity-Relationship-Diagramm		Erfedigt		
Tabellenstruktur erstellen		Erfedigt		
Dokumentation Projektbericht		Erfedigt		
Kurze Reflektion durch alle Teammitglieder	Lernrückblick dokumentieren	Erfedigt		
<b>Datenbanksysteme Woche 5</b>				
Kurze Reflektion durch alle Teammitglieder	Lernrückblick dokumentieren	Erfedigt		
Datenimport	Python	Erfedigt		
Datenbereinigung	Python	Erfedigt		
Datenaufbereitung	Python	Erfedigt		
Dokumentation Projektbericht		Erfedigt		
<b>Datenbanksysteme Woche 6</b>				
Kurze Reflektion durch alle Teammitglieder	Lernrückblick dokumentieren	Erfedigt		
Übungspräsentation		Erfedigt		
Datenintegration		Erfedigt		
Brainstorming User Stories		Erfedigt		
Brainstorming Use Cases		Erfedigt		
<b>Datenbanksysteme Woche 7</b>				
Kurze Reflektion durch alle Teammitglieder	Lernrückblick dokumentieren	Erfedigt		
Fehlende Daten holen	NPO und Natural Earth Countries	Erfedigt		
Datenbank neu strukturieren		Erfedigt		
Entity-Relationship-Diagramm anpassen		Erfedigt		
Tabellenstruktur anpassen		Erfedigt		
Arbeiten an dem Projektbericht		Erfedigt		
<b>Datenbanksysteme Woche 8</b>				
Kurze Reflektion durch alle Teammitglieder	Lernrückblick dokumentieren	Erfedigt		
VIEWS erstellt		Erfedigt		
Arbeiten an dem Projektbericht		Erfedigt		
<b>Datenbanksysteme Woche 9</b>				
Kurze Reflektion durch alle Teammitglieder	Lernrückblick dokumentieren	Erfedigt		
Erste Visualisierungen	Python Plotly	Erfedigt		
Arbeiten an dem Projektbericht		Erfedigt		
<b>Datenbanksysteme Woche 10</b>				
Kurze Reflektion durch alle Teammitglieder	Lernrückblick dokumentieren	Erfedigt		
Arbeiten an dem Projektbericht		Erfedigt		
Indexierung wegen Full scan		Erfedigt		
Types korrekt gewählt		Erfedigt		
<b>Datenbanksysteme Woche 11</b>				
Kurze Reflektion durch alle Teammitglieder	Lernrückblick dokumentieren	Erfedigt		
Sicherheit überprüft		Erfedigt		
Passwort verstärkt	Plugin (validate password)	Erfedigt		
Risiken angeschaut		Erfedigt		
Benutzer erstellt	developer, dashboard	Erfedigt		
Arbeiten an dem Projektbericht		Erfedigt		
<b>Datenbanksysteme Woche 12</b>				
Kurze Reflektion durch alle Teammitglieder	Lernrückblick dokumentieren	Erfedigt		
Dashboard erstellt	Python Streamlit	Erfedigt		
Arbeiten an dem Projektbericht		Erfedigt		
<b>Datenbanksysteme Woche 13</b>				
Kurze Reflektion durch alle Teammitglieder	Lernrückblick dokumentieren	Erfedigt		
Evaluationen der Visualisierungen		Erfedigt		
Arbeiten an dem Projektbericht		Erfedigt		
<b>Datenbanksysteme Woche 14</b>				
Kurze Reflektion durch alle Teammitglieder	Lernrückblick dokumentieren	Erfedigt		
Finalisierung des Projektberichtes		Erfedigt		
Durchlesen und eventuelle Korrekturen des Projektberichtes		Erfedigt		
Abschliessen des Projektberichtes		Erfedigt		
Projektpresentation		Erfedigt		
Dokument		Erfedigt	31-Dec-22	
Projektbericht Finalisieren				