Universität Rostock Traditio et Innovatio

Bachelorarbeit

Vorbereitung eines sedimentologischen Datensatzes zur Integration von Provenance

eingereicht von: Kristi Bartelt

eingereicht am: 22.03.2022

Gutachter: M.Sc. Tanja Auge
Dr.-Ing. Holger Meyer

Betreuer: M.Sc. Tanja Auge
Dr. rer. nat. Susanne Feistel
Dipl.-Inf. Susanne Jürgensmann
Dr. Svenja Papenmeier

Zusammenfassung

Am Leibniz-Institut für Ostseeforschung Warnemünde (IOW) werden auf Forschungsfahrten Sedimentproben zur sedimentologischen Kartierung der Ostsee entnommen. Die Sedimentproben werden im Labor mit verschiedenen Messmethoden auf die Verteilung der enthaltenen Korngrößen untersucht. Die Ergebnisse dieser Untersuchungen werden bisher dezentral in Dateien verschiedener Formate und Strukturen lokal auf den Rechnern der Wissenschaftler gespeichert. In dieser Arbeit werden die vorhandenen Dateistrukturen analysiert und ein für alle Strukturen geeignetes relationales Datenbankmodell entworfen. Zur besseren Rückverfolgbarkeit der Daten wird das Modell um zusätzliche Metadaten erweitert und durch Einfügen von global eindeutigen Tupel-IDs auf die Umsetzung von Data Provenance vorbereitet. Das erweiterte Modell wird unter Verwendung von MySQL implementiert. Im Anschluss wird beschrieben, wie der Import der Daten erfolgt und welche Vorverarbeitungsschritte die Daten dazu durchlaufen. Es werden exemplarische Datenbankanfragen gestellt, um nachzuweisen, dass das Konzept funktioniert. Abschließend wird in einem kurzen Fazit die Arbeit zusammengefasst und ein Ausblick auf weiterführende Themen gegeben.

Abstract

At the Leibniz Institute for Baltic Sea Research Warnemünde (IOW), sediment samples are collected on research cruises to map the sediments in the Baltic Sea. The sediment samples are taken to the laboratory and analyzed using different methods to determine the distribution of grain sizes. Currently the analysis results are saved locally on the computers of the researchers, in files of different formats and structures. In this bachelor thesis, the different file structures are analyzed and a relational database model is developed to encompass all structures. For improved traceability of data, the database model is extended to include additional metadata and globally unique tuple identifiers to prepare for data provenance. The extended model is implemented using MySQL. Afterwards, the import of data is explained, including the necessary preprocessing of the data. Exemplary database queries are written as a proof of concept. Finally, a summary of the thesis and recommendations for further research are given.

Inhaltsverzeichnis

1 Einleitung

Diese Bachelorarbeit beschäftigt sich mit der Verwaltung von Sedimentdaten am Leibniz-Institut für Ostseeforschung Warnemünde (IOW). Zur sedimentologischen Kartierung des Meeresbodens der Ostsee werden auf Forschungsfahrten Sedimentproben entnommen und im Labor auf die Verteilung der enthaltenen Korngrößen untersucht. Durch die Verwendung verschiedener Messverfahren liegen die Ergebnisse in unterschiedlichen Dateiformaten und -strukturen vor, die zudem dezentral gespeichert werden.

1.1 Problemstellung

In der Sektion der Marinen Geologie des IOW werden seit Jahrzehnten auf Forschungsfahrten verschiedene Daten gesammelt. Dazu zählen unter anderem *hydroakustische* Daten über die Tiefe und Beschaffenheit des Meeresbodens (d.h. Wasserschall-Messungen durch Sonare, Echolote o.ä.), Unterwasservideos und Analysen von *Sedimentproben*. Für letzteres werden kleine Mengen des Meeresbodens entnommen, der z.B. aus Sand, Kies oder Schlick besteht, und im Labor weiter analysiert. Diese Arbeit beschäftigt sich mit der Speicherung der Sedimentanalysen. Bei der Analyse wird die *Korngrößenverteilung* der Probe ermittelt, d.h. wie hoch der Anteil von kleinen und großen Körnern ist. Der Anteil der einzelnen Korngrößen, also der Durchmesser der einzelnen Sedimentkörner, wird in verschiedenen Korngrößenklassen festgehalten und statistisch ausgewertet. Bisher werden die Ergebnisse der Untersuchungen dieser Proben dezentral und in verschiedenen Strukturen und Formaten gespeichert.

Provenance bezeichnet im Allgemeinen Informationen über den Entstehungsprozess eines Endproduktes. Bei dem Endprodukt kann es sich um ein physisches Objekt oder um Daten handeln [HDB17]. Im Kontext der Datenverwaltung geht es jedoch um die Provenance von Daten jeglicher Art. Provenance soll die Rückverfolgbarkeit der Daten zu ihrem Ursprung ermöglichen. Dies dient der Qualitätskontrolle, Reproduzierbarkeit und der Stärkung von Vertrauen in die Daten. Mithilfe von Provenance können beispielsweise Forschungsergebnisse reproduziert und überprüft werden und eventuell vorhandene Fehler entdeckt und korrigiert werden.

Das Ziel dieser Arbeit besteht darin, die unterschiedlichen Strukturen der vorhandenen sedimentologischen Daten zu analysieren und einen geeigneten Datenbankentwurf zu entwickeln, in dem alle vohandenen Sedimentdaten gespeichert werden können. Dabei soll die Datenbank zur besseren Rückverfolgbarkeit von Forschungsergebnissen so erweitert werden, dass *Data* und/oder *Workflow Provenance* umgesetzt werden können. Data Provenance ermöglicht die Rückverfolgbarkeit von Ergebnissen einer Datenbankanfrage innerhalb der Datenbank. Metadata Provenance ermöglicht die Rückverfolgbarkeit des Weges von Rohdaten zu Forschungsergebnissen. Um das Ziel der Arbeit zu erreichen, werden die einzelnen Speicherformate der Daten getrennt voneinander untersucht und für jedes Speicherformat ein eigenes Entity-Relationship-Modell erstellt, welches die Struktur und Zusammenhänge der Daten auf einer abstrakten, implementierungsunabhängigen Ebene beschreibt. Die Modelle werden anschließend zu einem Gesamtmodell zusammengeführt. Das Gesamtmodell wird durch Tupel-IDs und Metadaten erweitert,

um die Anwendung von Provenance-Verfahren zu ermöglichen. Aus dem um Provenancetechniken erweiterten ER-Modell wird ein Relationenmodell erstellt. Auf Basis des Relationenmodells kann dann eine prototypische Implementierung der entworfenen Datenbank erfolgen und die Daten eingepflegt werden.

1.2 Leibniz-Institut für Ostseeforschung Warnemünde

Das Leibniz-Institut für Ostseeforschung Warnemünde[1] ist eine außeruniversitäre Forschungseinrichtung zur Untersuchung von Küsten- und Randmeeren, insbesondere der Ostsee. Es wurde unter seinem heutigen Namen 1992 gegründet und ging aus dem ehemaligen Institut für Meereskunde Warnemünde (IfM) der DDR hervor [Fen18].

Im aktuellen Forschungsprogramm des IOW [Leia] wird der besondere Fokus auf Küsten- und Randmeere betont, da diese im Vergleich zum offenen Meer Orte von biologischer Hochproduktion darstellen und besonders bedeutend für die Stoffkreisläufe unseres Planeten sind. Aufgrund der Nähe des Menschen ist auch der anthropogene Einfluss auf die Bedingungen in Küsten- und Randmeeren größer als im offenen Meer. Der zentrale Auftrag des Forschungsprogramms lautet daher: „Verstehen, wie sich die Systeme vor dem Hintergrund des Klimawandels und einer intensiven Nutzung durch den Menschen verändern“ [Leia].

Die Forschung des IOW zeichnet sich durch die interdisziplinare Zusammenarbeit seiner vier Sektionen aus: Marine Geologie, Meereschemie, Biologische Meereskunde und Physikalische Ozeanographie. Die Sektion Marine Geologie beschäftigt sich mit den geologischen und geochemischen Komponenten der Ostsee. In der Sektion Meereschemie werden raum-zeitliche Veränderungen von Wasserinhalten wie Gasen oder Nährstoffen untersucht. Die Sektion Biologische Meereskunde erforscht die Organismen im Ökosystem der Ostsee und deren Stoffkreisläufe. Die Sektion Physikalische Ozeanographie verfolgt das Ziel, das Verständnis physikalischer Prozesse und der Dynamik mariner Ökosysteme weiter voranzutreiben [Leib].

Die Forschungsdaten, mit denen sich diese Arbeit beschäftigt, stammen aus der Arbeitsgruppe Marine Geophysik der Sektion Marine Geologie. Ein Schwerpunkt der Arbeitsgruppe ist die Sedimentkartierung. In diesem Rahmen werden auf Forschungsfahrten hydroakustische Messsysteme genutzt, die durch Unterwasserschallwellen Daten über die Tiefe und Beschaffenheit des Meeresbodens sammeln. Zur Validierung der hydroakustisch erfassten Daten werden mithilfe von Greifern Sedimentproben entnommen. Diese Sedimentproben werden im Labor auf verschiedene Parameter untersucht, wobei der Fokus auf der Analyse der Korngrößenverteilung liegt. Es stehen verschiedene Methoden zur Messung der Korngrößenverteilung zur Verfügung, deren Ergebnisse in unterschiedlichen Strukturen und Dateiformaten gespeichert werden. Hintergründe zu den Messverfahren, den resultierenden Datenformaten und der Datengrundlage für diese Arbeit sind in Kapitel 4.1 zu finden.

1.3 Aufbau der Arbeit

Kapitel 2 behandelt die theoretischen Grundlagen, die im weiteren Verlauf der Arbeit benötigt werden. Dazu gehören das Entity-Relationship-Modell, das Relationenmodell, die Datendefinitionssprache und Provenance.

[1] `https://io-warnemuende.de/`

In Kapitel 3 wird der aktuelle Stand der Forschung und der Technik beschrieben. Zunächst werden einige wissenschaftliche Arbeiten vorgestellt, die sich mit verwandten Themen beschäftigen. Danach erfolgt ein Überblick über das Datenbankmanagementsystem MySQL, welches für die Implementierung der Datenbank genutzt wird.

Kapitel 4 beschreibt das Konzept zur Lösung der Problemstellung. Es werden die Daten vorgestellt, auf denen diese Arbeit basiert, ein Entity-Relationship erstellt und um Provenance erweitert, basierend darauf ein Relationenmodell erstellt und die Vorbereitung der Daten zum Import in die Datenbank erläutert.

In Kapitel 5 wird das Relationenmodell mit dem Datenbankmanagementsystem MySQL implementiert. Danach werden die Beispieldatensätze für den Import in die Datenbank vorbereitet und importiert.

Kapitel 6 bildet den Abschluss der Arbeit. Es erfolgt eine Zusammenfassung der Arbeit und ein Ausblick auf mögliche weiterführende Themen, die im Rahmen dieser Arbeit nicht behandelt werden konnten.

2 Grundlagen

In diesem Kapitel werden grundlegende Begriffsdefinitionen eingeführt und Konzepte erläutert, die im weiteren Verlauf der Arbeit genutzt werden. Dabei handelt es sich um die Datenbankenkonzepte Entity-Relationship-Modell und Relationenmodell, die Abbildung des ER-Modells auf das Relationenmodell und die Datendefinitionssprache SQL. Danach wird auf die Grundlagen von Provenance eingegangen.

2.1 Entity-Relationship-Modell

In diesem Abschnitt wird das *Entity-Relationship-Modell* oder ER-Modell eingeführt. Es handelt sich dabei um ein grundlegendes Modell der frühen Phase des Datenbankentwurfs. Das ER-Modell beschreibt die Struktur der Datenbank auf einer abstrakten, implementierungsunabhängigen Ebene durch eine graphische Notation. Im Folgenden werden einige Begriffe definiert, auf denen das ER-Modell basiert.

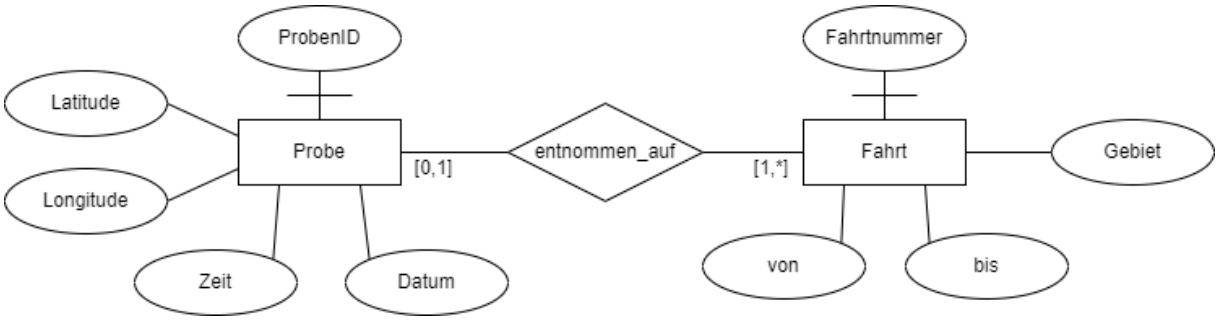

Abbildung 2.1: Beispiel für ein ER-Modell

Definition 2.1 (Entity-Typen, Beziehungstypen, Attribute [HSS18])**.**
Entity-Typen: Entitys sind die in einer Datenbank zu repräsentierenden Informationseinheiten. Sie werden nicht einzeln in ihren Ausprägungen (eine konkrete Probe mit der ProbenID „AT2020_10“, die am 09.07.2020 um 10:42 Uhr entnommen wurde) modelliert. Stattdessen werden Objekte mit gleichen Eigenschaften zu Entity-Typen zusammengefasst wie beispielsweise Probe. In der graphischen Notation werden Entity-Typen als Rechtecke dargestellt, wobei der Name des Typs in das Rechteck eingetragen wird. In Abbildung 2.1 sind die Entity-Typen `Probe` und `Fahrt` dargestellt.

Beziehungstypen: Auch Beziehungen zwischen Entitys werden im ER-Modell nicht für jede Ausprägung modelliert, sondern zu Beziehungstypen zusammengefasst. Es wird also nur der Sachverhalt beschrieben, dass eine Probe auf einer Fahrt entnommen wurde und nicht, dass die Probe mit der ProbenID „AT2020_10“ auf der Forschungsfahrt mit der Fahrtnummer „AT2020“ entnommen wurde. Beziehungstypen werden graphisch als eine Raute dargestellt, in die der Name des Typs eingetragen wird. Die Verbindungen zu den beteiligten Entity-Typen erfolgt über Kanten typischerweise von den Ecken der Raute. In Abbildung 2.1 ist der Beziehungstyp `entnommen_auf` zwischen den Entity-Typen `Probe` und `Fahrt` dargestellt.

Attribute: Attribute modellieren Eigenschaften von Entitys oder auch Beziehungen. Alle Entitys eines Entity-Typs haben dieselben Arten von Eigenschaften. Attribute werden somit für Entity-Typen deklariert. In der graphischen Notation wird ein Attribut als Oval dargestellt und über eine Kante dem Entity-Typ zugeordnet. In Abbildung 2.1 ist beispielsweise der Entity-Typ `Probe` mit den Attributen `ProbenID, Latitude, Longitude, Zeit` und `Datum` dargestellt.

Definition 2.2 (Schlüssel [HSS18])**.**
Schlüssel: In der Regel haben wir bei einer Beschreibung im ER-Modell den Effekt, dass einige Attribute mit ihren Werten bereits eine eindeutige Identifizierung für Entitys eines Entity-Typs bilden. Wir bezeichnen eine derartige Identifikation durch eine Attributmenge als Schlüssel für den betreffenden Entity-Typ. In jedem Datenbankzustand identifizieren die aktuellen Werte der Schlüsselattribute eindeutig Instanzen des Entity-Typs. In der graphischen Notation werden Schlüssel dargestellt, indem die Kanten zwischen Entity und Schlüsselattributen einen Querbalken erhalten. In Abbildung 2.1 ist der Entity-Typ `Probe` mit dem Schlüssel `ProbenID` dargestellt.

Definition 2.3 (Kardinalitäten [HSS18])**.**
Kardinalitäten: Neben der Anzahl der in eine Beziehung eingehenden Entity-Typen ist die Anzahl der beteiligten Instanzen des jeweiligen Entity-Typs ein weiteres wichtiges Merkmal von Beziehungstypen. Derartige Festlegungen bezeichnet man als Kardinalität einer Beziehung. Eine Kardinalitätsangabe stellt im Prinzip eine Integritätsbedingung dar. D.h. damit wird beschrieben, unter welchen Bedingungen die Daten konsistent zur realen Welt sind. Allgemein lassen sich drei Formen von Kardinalitäten unterscheiden, wobei m und n für eine beliebige Anzahl > 1 stehen:

- 1:1-Beziehungen,
- 1:n- bzw. n:1-Beziehungen sowie
- n:m-Beziehungen.

Es gibt verschiedene Notationen zur graphischen Darstellung von Kardinalitäten. In dieser Arbeit wird die Intervallnotation verwendet, bei der zu jedem an der Beziehung beteiligten Entity-Typ der minimale und der maximale Wert der möglichen Teilnahmen der Instanzen vorgegeben wird. Die Kardinalitätsangaben werden an der Verbindungslinie zum jeweiligen Entity-Typ notiert und sind somit auch für beliebige $n-$stellige Beziehungstypen verwendbar. In Abbildung 2.1 hat die Beziehung `entnommen_auf` eine Kardinalität von 1:n bzw. [1,*]:[0,1]. Das heißt, einer Instanz des Entity-Typs `Probe` kann eine oder keine Instanz des Entity-Typs `Fahrt` zugeordnet werden, während einer Instanz des Entity-Typs `Fahrt` mindestens eine und maximal unendlich viele Instanzen des Entity-Typs `Probe` zugeordnet werden können.

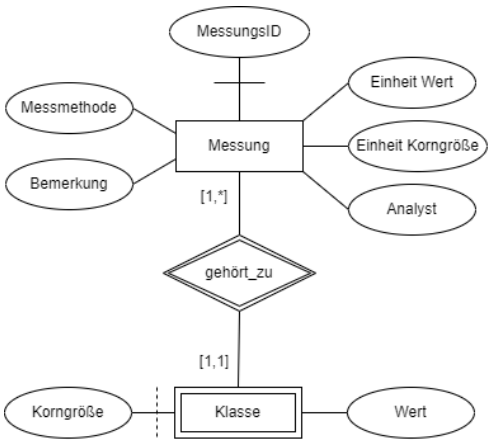

Abbildung 2.2: Beispiel für ein ER-Modell mit einem abhängigen Entity-Typen

Definition 2.4 (Funktionale Beziehungen, Abhängige Entity-Typen [HSS18])**.**
Funktionale Beziehungen: Einen Beziehungstyp mit der Kardinalität n:1 bezeichnet man auch als funktionale Beziehung.

Abhängige Entity-Typen: Abhängige Entity-Typen sind Entity-Typen, an deren Schlüssel eine funktionale Beziehung beteiligt ist. Instanzen eines derartig abhängigen Entity-Typs können somit nur existieren in Abhängigkeit von anderen Entitys. In der graphischen Notation ist dies durch eine doppelte Umrandung des abhängigen Entity-Typen und des betroffenen Beziehungstyps gekennzeichnet. In Abbildung 2.2 ist der von `Messung` abhängige Entity-Typ `Klasse` zu sehen, an dessen Schlüssel die funktionale Beziehung `gehört_zu` beteiligt ist.

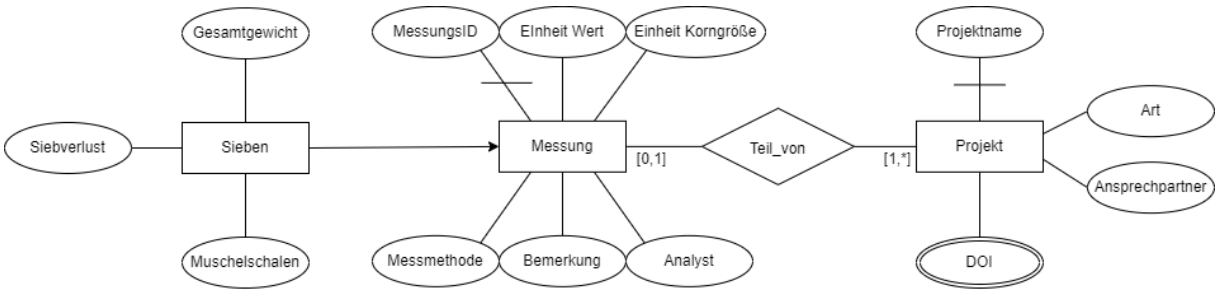

Abbildung 2.3: Beispiel für ein ER-Modell mit einer Ist-Beziehung und mengenwertigem Attribut

Definition 2.5 (Ist-Beziehungen, mengenwertige Attribute [HSS18])**.**
Ist-Beziehungen: Bei der Ist-Beziehung handelt es sich um eine Spezialisierungs-/Generalisierungsbeziehung und einen Spezialfall eines abhängigen Entity-Typs. Ein Entity-Typ E1 IST ein Entity-Typ E2, wenn es sich bei E1 um eine Spezialisierung von E2 handelt. In der graphischen Notation wird die Ist-Beziehung nicht wie eine normale Beziehung durch eine Raute dargestellt, sondern durch einen Pfeil vom spezielleren Entity-Typen zum genelleren Entity-Typen. In Abbildung 2.3 ist der Entity-Typ `Sieben` eine Spezialisierung vom Entity-Typ `Messung`.

Mengenwertige Attribute: Ein mengenwertiges Attribut ist ein Attribut, bei dem einem Tupel eine Menge von Werten als Attributwert zugeordnet wird. Das heißt, ein Attributwert ist beispielsweise ein Array oder eine Liste von Werten. In der graphischen Notation werden mengenwertige Attribute doppelt umrandet. In Abbildung 2.3 ist das Attribut `DOI` des Entity-Typen `Projekt` ein mengenwertiges Attribut, in dem mehrere DOIs pro Tupel gespeichert werden können.

Wie bereits erwähnt, beschreibt das Entity-Relationship-Modell den Datenbankentwurf auf eine abstrakte, implementierungsunabhängige Weise. Dies ist vorteilhaft für die frühe Phase des Datenbankentwurfs, da das ER-Modell auch für fachfremde Personen einfach verständlich ist [HSS18]. Als Basis für die Implementierung des Datenbankentwurfs ist jedoch ein implementierungsnäheres Modell von Vorteil. Für die Implementierung einer relationalen Datenbank ist daher das Relationenmodell besonders geeignet, welches im nächsten Abschnitt vorgestellt wird.

2.2 Relationenmodell

In diesem Abschnitt wird das *Relationenmodell* eingeführt. Es handelt sich dabei um das am weitesten verbreitete Datenbankmodell, da die kommerziell erfolgreichsten Datenbanksysteme relationale Systeme sind [HSS18]. Dabei werden Daten in Tabellen, sogenannten Relationen, gespeichert. Im Folgenden werden einige Begriffe definiert, auf denen das Relationenmodell basiert.

Messung

MessungsID	Einheit Wert	Einheit Korngröße	Analyst	Messmethode	Bemerkung
1	prozent	mikrometer	Gründling	Mastersizer wet	
2	prozent	mikrometer	messlab	Mastersizer wet	Average of AT2020_23
3	gramm	millimeter		Sieben	2. mal gesiebt von Svenja

Klasse

MessungsID	Korngröße	Wert
1	51.83	0.07
1	58.88	0.54
2	0.41	0
2	0.46	0.05
3	16	10.21
3	13.2	0

Tabelle 2.1: Beispielrelationen für die Entity-Typen `Messung` und `Klasse` aus Abbildung 2.2

Definition 2.6 (Relation und Relationenschema [HSS18])**.**
Relation: Eine gespeicherte Tabelle wird als Relation bezeichnet. Der Relationenname steht der tabellarischen Darstellung voran. Die Begriffe Relationenname und Relation werden oft synonym genutzt.

Relationenschema: Der „Tabellenkopf“ legt die Anzahl, Bezeichnung und Typisierung der Spalten fest. Er wird als Relationenschema bezeichnet.

Definition 2.7 (Attribut, Tupel, Attributwert [HSS18])**.**
Attribut: Eine Spaltenüberschrift im Tabellenkopf legt ein Attribut fest.

Tupel: Eine Zeile der Tabelle bezeichnen wir als Tupel. Die Menge aller Einträge, also aller gespeicherten Tupel, bildet die Relation an sich.

Attributwert: Ein Eintrag in ein Feld eines Tupels ist ein konkreter Attributwert.

Definition 2.8 (Schlüssel, Primärschlüssel, Fremdschlüssel, Surrogatschlüssel)**.**
Schlüssel: Ein Schlüssel einer Relation ist ein Attribut oder eine Menge von Attributen, die jedes Tupel der Relation eindeutig identifiziert.

Primärschlüssel: Eine Relation kann mehrere Schlüssel enthalten. Dann wird einer der Schlüssel als Primärschlüssel ausgewählt und in der graphischen Darstellung unterstrichen.

Fremdschlüssel: Eine Relation kann einen Schlüssel einer anderen Relation enthalten, dieser wird dann als Fremdschlüssel bezeichnet. Fremdschlüssel dienen als eindeutige Verweise auf andere Relationen.

Surrogatschlüssel: Ein Surrogatschlüssel ist ein Schlüssel, der künstlich in eine Relation eingefügt wurde. Er enthält keine Nutzdaten und dient nur der Identifikation von Tupeln innerhalb der Relation. Surrogatschlüssel werden beispielsweise eingesetzt, wenn eine Relation keine Attributmenge enthält, die als Schlüssel geeignet ist, oder vorhandene Schlüssel sehr komplex sind.

Definition 2.9 (Datenbank und Datenbankschema [HSS18])**.**
Datenbank: Eine Datenbank ist eine Menge von Relationen.

Datenbankschema: Ein Datenbankschema ist eine Menge von Relationenschemata.

Da der Datenbankentwurf in der Regel mit dem Entwurf eines ER-Modells beginnt und das Relationenmodell besser als Basis zur Implementierung einer relationalen Datenbank geeignet ist, muss das ER-Modell in ein Relationenmodell überführt werden. Hierzu gibt es Abbildungsregeln, die sicherstellen, dass das Relationenmodell die gleichen Informationen enthält wie das ER-Modell. Dieser Vorgang wird im nächsten Abschnitt vorgestellt.

2.3 Abbildung vom ER-Modell auf das Relationenmodell

In diesem Abschnitt wird erläutert, wie ein ER-Modell in ein Relationenmodell überführt wird. Dieser Vorgang folgt klar definierten Regeln, um die Qualität des erstellten Relationenmodells zu gewährleisten. Alle Informationen, die im ER-Modell enthalten sind, müssen auch im Relationenmodell dargestellt sein. In den folgenden Abschnitten werden die Abbildungsregeln für die Überführung von ER-Modellen in Relationenmodelle vorgestellt und einige Details der Regeln genauer betrachtet.

2.3.1 Abbildungsregeln

Die hier aufgeführten Abbildungsregeln stammen aus [HSS18]. Sie fassen die grundlegenden Schritte der Abbildung vom ER-Modell auf das Relationenmodell zusammen.

- Entity-Typen und Beziehungstypen werden jeweils auf Relationenschemata abgebildet.
- Bei Entity-Typen werden die Attribute zu Attributen des Relationenschemas, die Schlüssel werden übernommen. Bei Beziehungstypen werden ebenso die Attribute (falls vorhanden) in das korrespondierende Relationenschema übernommen, dazu kommen dann noch die Primärschlüssel aller an dieser Beziehung beteiligten Entity-Typen.
- Die verschiedenen Kardinalitäten der Beziehungen werden durch Wahl der Schlüssel bei dem zugehörigen Relationenschema ausgedrückt.
- In einigen Fällen können Relationenschemata von Entity- und Beziehungstypen miteinander verschmolzen werden.
- Zwischen den verbleibenden Relationenschemata werden diverse Fremdschlüsselbedingungen eingeführt.

2.3.2 Wahl der Primärschlüssel

Bei der Abbildung auf das Relationenmodell ist besonders auf die *Informationskapazität* der Modelle zu achten. Das Ziel ist die Erhaltung der Informationskapazität, das heißt im Relationenmodell sind genauso viele Instanzen darstellbar wie im ER-Modell. Wenn im Relationenmodell mehr Instanzen darstellbar sind spricht man von einer Erhöhung der Informationskapazität, wenn weniger Instanzen darstellbar sind spricht man von einer Verminderung der Informationskapazität. Die Informationskapazität eines Relationenmodells ist von der Wahl der Primär- und Fremdschlüssel abhängig [HSS18].

Entity-Typen Enthält ein Entity-Typ mehrere Schlüssel, wird im Relationenschema einer davon als Primärschlüssel gewählt. Dieser sollte möglichst einfach sein, d.h. im Idealfall sollte der Primärschlüssel aus einem einzelnen integer-Attribut bestehen statt aus mehreren Attributen oder komplexeren Datentypen. Die anderen Schlüsselkandidaten können während der Datendefinition mit der unique-Klausel als Schlüssel spezifiziert werden [HSS18].

Beziehungstypen Bei Beziehungstypen basiert die Wahl des Primärschlüssels auf den Kardinalitäten der Beziehung. Handelt es sich um eine n:m- bzw. [_,*]:[_,*]-Beziehung, werden die Primärschlüssel beider beteiligten Entity-Typen gemeinsam zum Primärschlüssel des Relationenschemas. Für 1:n- bzw. [_,*]:[_,1]-Beziehungen wird der Primärschlüssel der Entity auf der n- bzw. [_,1]-Seite zum Primärschlüssel des Relationenschemas. Bei 1:1- bzw [_,1]:[_,1]-Beziehungen wird einer der beiden Schlüssel zum Primärschlüssel gewählt, der andere Schlüssel wird während der Datendefinition als unique spezifiziert [HSS18].

2.3.3 Verschmelzen von Relationenschemata

In manchen Fällen können die Relationenschemata von Beziehungstypen und an ihnen beteiligten Entity-Typen miteinander verschmolzen werden, d.h. zwei oder mehr Relationenschemata werden zu einem Schema zusammengefügt, das alle Attribute der zu verschmelzenden Schemata enthält. Verschmelzungen sind nicht möglich, wenn es sich um einen optionalen Beziehungstyp handelt, an dem Instanzen eines beteiligten Entity-Typs teilnehmen können, aber nicht müssen. Ein optionaler Beziehungstyp wird also durch Kardinalitäten [0,1] oder [0,*] beschrieben. Bei nicht-optionalen bzw. zwingenden Beziehungstypen mit Kardinalitäten [1,1] oder [1,*], können folgende Verschmelzungen vorgenommen werden:

- Handelt es sich um eine 1:n- bzw. [1,*]:[1,1]-Beziehung, können die Relationenschemata des Beziehungstyps und des Entity-Typs auf der [1,1]-Seite miteinander verschmolzen werden.

- Handelt es sich um eine 1:1- bzw. [1,1]:[1,1]-Beziehung, können die Relationenschemata des Beziehungstyps und beider Entity-Typen miteinander verschmolzen werden.

Verschmelzungen von optionalen Beziehungstypen können nicht vorgenommen werden, da dies zu Nullwerten in Schlüsseln und Fremdschlüsseln führen könnte. Verschmelzungen auf der [1,*]-Seite von zwingenden Beziehungstypen würden hingegen zu Redundanzen führen. Daher werden Verschmelzungen nur auf [1,1]-Seiten von Beziehungen vorgenommen [HSS18].

2.3.4 Ist-Beziehung

Für die Ist-Beziehung wird kein eigenes Relationenschema erstellt. Das Relationenschema des spezielleren Entity-Typs erbt jedoch den Primärschlüssel des allgemeineren Entity-Typs. Somit ist der Primärschlüssel des speziellen Entity-Typs auch Fremdschlüssel aus dem allgemeinen Entity-Typ. Daher können in der spezielleren Relation nur Schlüsselwerte existieren, die auch in der allgemeineren Relation vorhanden sind [HSS18].

2.3.5 Normalformen

Nach Ermittlung der grundlegenden Relationenschemata, werden diese auf Erfüllung von Normalformen überprüft und ggf. angepasst. Die Normalformen dienem dem Zweck der Entfernung von Redundanzen aus den Schemata, um Verschwendung von Speicherplatz und sogenannte Änderungsanomalien zu vermeiden. Im Folgenden werden die einzelnen Normalformen und ihre Anforderungen erläuert.

Erste Normalform Die erste Normalform oder 1NF erlaubt nur atomare Attribute in den Relationenschemata. Das heißt, Attributwerte dürfen nur Elemente von einfachen Datentypen wie String oder Integer sein, jedoch keine mengenwertige Attribute mit Datentypen wie Arrays oder Sets. Sind solche mengenwertigen Attribute vorhanden, müssen betroffene Tupel so oft dupliziert werden, dass für jedes Element des mengenwertigen Attributs ein eigenes Tupel existiert und alle Attribute atomar sind. Dies erzeugt zwar Redundanzen, die aber in den weiteren Normalformen eliminiert werden [HSS18].

Zweite Normalform Die zweite Normalform oder 2NF erlaubt keine partiellen Abhängigkeiten zwischen den Schlüsseln des Relationenschemas und weiteren Attributen, die keine Primattribute sind. Ein Primattribut ist ein Attribut, das Teil eines Schlüssels der Relation ist. Eine partielle Abhängigkeit liegt vor, wenn eine Attributmenge funktional von einem Teil des Schlüssels abhängt. Ist dies der Fall, dann eliminiert man die rechte Seite der partiellen Abhängigkeit und verschiebt sie zusammen mit der linken Seite in ein neues Relationenschema [HSS18].

Dritte Normalform Die dritte Normalform oder 3NF erlaubt keine transitiven Abhängigkeiten von Schlüsseln im Relationenschema. Eine transitive Abhängigkeit $K \to X \to A$ liegt vor, wenn eine Attributmenge X funktional von der Attributmenge K abhängig ist und eine weitere Attributmenge A funktional von X abhängig ist. Dabei werden nur transitiv abhängige Attribute untersucht, die keine Primattribute sind. Ist in einem Relationenschema eine transitive Abhängigkeit vorhanden, so wird die transitiv abhängige Attributmenge A in ein neues Relationenschema verschoben und die zwischengeschaltete Attributmenge X zusätzlich in das neue Schema kopiert [HSS18].

Bisher wurden die theoretischen Grundlagen des ER-Modells, des Relationenmodells und das Verfahren zum Abbilden eines ER-Modells auf ein Relationenmodell eingeführt. Mit diesen Grundlagen kann eine relationale Datenbank entworfen werden, es fehlen jedoch noch die nötigen Grundlagen, mit denen der Datenbankentwurf implementiert werden kann. Hierzu wird im nächsten Abschnitt die Datendefinitionssprache von SQL vorgestellt.

2.4 Datendefinitionssprache

Im Relationenmodell werden die Relationenschemata der Datenbank eher abstrakt modelliert. Um die Datenbank zu implementieren, wird eine *Datendefinitionssprache* (DDL) benötigt. Mithilfe einer DDL werden die Relationenschemata so definiert, dass sie von einem Datenbankmanagementsystem (DBMS) erstellt werden können. Dazu müssen unter anderem auch die Datentypen der Spalten sowie Schlüssel- und Fremdschlüsselbedingungen definiert werden. Im Rahmen dieser Bachelorarbeit werden das DBMS MySQL und die dazugehörige Datenbanksprache SQL genutzt. Bei SQL handelt es sich um eine universelle Datenbanksprache, die mehrere Teile umfasst. Dazu gehören die Datendefinitionssprache, Sichtdefinitionssprache, Anfragesprache, Datenänderungssprache und weitere Teilsprachen. In diesem Abschnitt werden die Grundbefehle SQLs kurz erläutert, die zur Datendefinitionssprache gehören. Die hier verwendeten Anweisungen zur Veranschaulichung der Syntax von Datendefinitionsanweisungen basieren auf [HSS18], wurden jedoch um zusätzliche Informationen aus [Ora] erweitert.

Relationen erstellen Um eine neue Relation zu erstellen, wird die Anweisung CREATE TABLE genutzt. Dabei wird erst der Relationenname angegeben, gefolgt von einer oder mehreren Spalten. Jede Spalte stellt ein Attribut dar und enthält den Spaltennamen, den Wertebereich des Attributs in Form eines SQL-Datentyps und wenn nötig können optional weitere Optionen definiert werden. Dazu gehören beispielsweise NOT NULL wenn keine leeren Attributwerte erlaubt werden, PRIMARY KEY wenn die

Spalte der Primärschlüssel der Relation ist, UNIQUE wenn die Spalte ein Schlüssel der Relation ist oder DEFAULT um einen Defaultwert anzugegeben. Außerdem können optional weitere Integritätsbedingungen wie Schlüssel aus mehreren Spalten oder Fremdschlüssel angegeben werden. Anweisung 2.1 zeigt die allgemeine Syntax einer CREATE-TABLE-Anweisung.

```
CREATE TABLE relationenname (
        spaltenname1 wertebereich1 [optionen],
        ...
        spaltennamen wertebereichn [optionen],
        [bedingungen]
)
```

Anweisung 2.1: Syntax einer SQL-Anweisung zum Erstellen einer neuen Relation

Relationen löschen Um eine Relation zu löschen, wird die Anweisung DROP TABLE genutzt. Dabei wird der Relationenname angegeben und optional kann der gewünschte Modus der Löschung angegeben werden. Standardmäßig ist der Löschmodus auf „restrict“ festgelegt. Im „restrict“-Modus wird die DROP-Anweisung nicht durchgeführt, wenn noch Sichten oder Integritätsbedingungen existieren, die zu der zu löschenden Relation gehören. Alternativ kann der Modus „cascade“ ausgewählt werden, bei dem die Löschung erzwungen wird, indem alle zugehörigen Sichten und Integritätsbedingungen gelöscht werden. Anweisung 2.2 zeigt die allgemeine Syntax einer DROP-TABLE-Anweisung.

```
DROP TABLE relationenname [option]
```

Anweisung 2.2: Syntax einer SQL-Anweisung zum Löschen einer Relation

Relationen ändern Um eine Relation zu ändern, wird die Anweisung ALTER TABLE genutzt. Dabei wird erst der Relationenname angegeben, gefolgt von der Modifikation, die an der Relation vorgenommen werden sollen. Zu den möglichen Modifikationen gehören:

- ADD COLUMN fügt eine neue Spalte in die Relation ein
- ADD CONSTRAINT fügt Integritätsbedingungen wie Schlüsselbedingungen in die Relation ein
- DROP COLUMN löscht eine Spalte aus der Relation
- DROP CONSTRAINT löscht Integritätsbedingungen aus der Relation
- ALTER COLUMN ändert den Defaultwert der Spalte

Anweisung 2.3 zeigt die allgemeine Syntax einer ALTER-TABLE-Anweisung, sowie die Syntax der oben aufgeführten Modifikationen.

Nachdem die grundlegenden Konzepte des Datenbankentwurfs und der Datendefinitionssprache eingeführt wurden, behandelt der folgende Abschnitt das Thema Provenance. Hierzu wird der Begriff Provenance erläutert und die Formen der Provenance vorgestellt.

```
ALTER TABLE relationenname modifikation

mögliche Modifikationen:
        ADD COLUMN spaltenname wertebereich [optionen]
        ADD CONSTRAINT [bedingungsname] bedingung
        DROP COLUMN spaltenname
        DROP CONSTRAINT bedingungsname
        ALTER COLUMN spaltenname SET DEFAULT defaultwert
```

Anweisung 2.3: Syntax einer SQL-Anweisung zum Ändern einer Relation, darunter die Syntax der möglichen Änderungsmöglichkeiten

2.5 Provenance

Der Begriff Provenance im Allgemeinen bezeichnet Informationen über den Entstehungsprozess eines Endproduktes. Dabei kann es sich sowohl um physische Objekte als auch um Daten handeln [HDB17]. Provenance soll dabei die Rückverfolgbarkeit des Produktes zu seinem Ursprung ermöglichen. Dies dient dem Zweck der Qualitätskontrolle, Reproduzierbarkeit und um Vertrauen in das Endprodukt zu verstärken.

Im Kontext der Informatik und Datenverwaltung ist die Provenance von Daten jeglicher Art von Bedeutung. Der Begriff Provenance ist jedoch breit gefächert und enthält verschiedene Arten mit eigenen Techniken und Zielen. So kann je nach verwendeter Technik z.B. der Ursprung von Ergebnissen einer Datenbankanfrage bestimmt werden, der Weg von Rohdaten zu Endergebnissen nachvollzogen werden oder Informationen über Änderungen an einer Datei anhand von Metadaten bestimmt werden.

In [HDB17] wird eine Hierarchie-Pyramide der Provenance-Arten eingeführt, zu sehen in Abbildung 2.4. Dabei werden zwei Eigenschaften betrachtet: die Allgemeinheit der gespeicherten Provenance-Informationen (vgl. Abbildung 2.4, Abnahme der Allgemeinheit in Richtung des blauen Pfeils) und das Level der Automatisierbarkeit, mit der die Provenance-Informationen erfasst werden können (vgl. Abbildung 2.4, Abnahme der Automatisierbarkeit in Richtung des orangenen Pfeils.) Die Basis der Pyramide bildet die sehr allgemeine und schwer automatisierbare Metadata Provenance, nach oben hin sinkt dann die Allgemeinheit und die Automatisierbarkeit steigt bis zur spezifischsten und automatisierbarsten Provenance-Art, der Data Provenance. In den folgenden Abschnitten werden drei Arten der Provenance genauer erläutert.

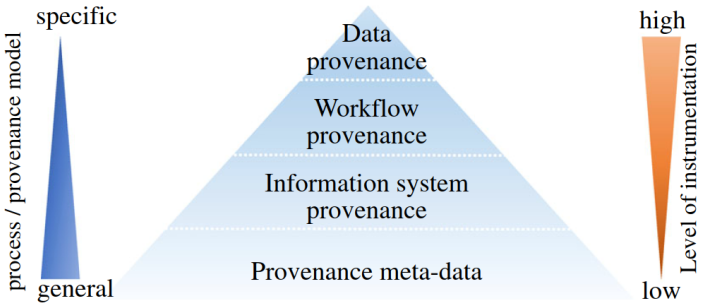

Abbildung 2.4: Hierarchie der Provenance-Arten nach [HDB17]

2.5.1 Data Provenance

Data Provenance legt den Fokus auf die Rückverfolgbarkeit der Ergebnisse von Datenbankanfragen. Es soll für alle Anfragen möglich sein, herauszufinden, aus welchen Relationen einzelne Tupel des Anfra-

```
SELECT Messung.Analyst FROM Messung
    INNER JOIN Klasse ON Messung.MessungsID = Klasse.MessungsID
    WHERE Klasse.Wert > 0 AND Klasse.Wert < 1
```

Anweisung 2.4: SQL-Anfrage auf die Beispieldaten aus Tabelle 2.2

geergebnisses stammen, wieso diese Tupel Teil des Ergebnisses sind, und wie das Ergebnis berechnet wurde. Da Data Provenance auf strukturierten Datenmodellen und Anfragesprachen mit klar definierter Semantik aufbaut, handelt es sich dabei um die spezifischste und am besten automatisierbare Art von Provenance.

Data Provenance kann mithilfe von Tupel-IDs ermöglicht werden, wobei jedem Tupel jeder Relation eine einzigartige ID zugewiesen wird. Formale Definitionen der Data Provenance sind in [CCT09] zu finden. Im Folgenden werden die drei Hauptfragen der Data Provenance anhand einer Datenbankanfrage auf Beispielen aus den Sedimentdaten erklärt. Die SQL-Anweisung 2.4 ist eine Datenbankanfrage auf die Daten in Tabelle 2.2 und erfragt alle Analysts, welche Messungen durchgeführt haben, die einen Messwert zwischen 0 und 1 enthalten. Tabelle 2.3 enthält das Ergebnis der Anfrage, sowie die Where-, Why- und How-Provenance der Ergebnistupel.

Messung

MessungsID	Einheit Wert	Einheit Korngröße	Analyst	Messmethode	Bemerkung	ProvID
1	prozent	mikrometer	Gründling	Mastersizer wet		m_1
2	prozent	mikrometer	messlab	Mastersizer wet	Average of AT2020_23	m_2
3	gramm	millimeter		Sieben	2. mal gesiebt von Svenja	m_3

Klasse

MessungsID	Korngröße	Wert	ProvID
1	51.83	0.07	k_1
1	58.88	0.54	k_2
2	0.41	0	k_3
2	0.46	0.05	k_4
3	16	10.21	k_5
3	13.2	0	k_6

Tabelle 2.2: Wiederholung von Beispielrelationen aus Tabelle 2.1 mit hinzugefügten Tupel-IDs

Analyst	Where			Why	How
Gründling	Messung	m_1	(Messung, m_1, Gründling)	$\{\{m_1, k_1\}, \{m_1, k_2\}\}$	$m_1 \cdot k_1 + m_1 \cdot k_2$
messlab	Messung	m_2	(Messung, m_2, messlab)	$\{\{m_2, k_4\}\}$	$m_2 \cdot k_4$

Tabelle 2.3: Anfrageergebnis der Anweisung 2.4 mit Data Provenance

Where-Provenance Where-Provenance beantwortet die Frage, aus welchen Relationen und/oder Tupeln der Ursprungsrelationen einzelne Tupel der Ergebnisrelation stammen. Sie kann in verschiedenen Formen angegeben werden: nur die Ursprungsrelation, nur die Ursprungstupel, oder als Kombination von Ursprungsrelation, -tupel und -wert. In Tabelle 2.3 sind alle drei Formen der Where-Provenance angegeben. Die erste Teilspalte gibt die Where-Provenance in Form der Ursprungsrelation an. Da beide Ergebnistupel aus der Relation `Messung` stammen, ist die Antwort für beide Tupel „Messung“. Die zweite Teilspalte gibt die Where-Provenance in Form des Ursprungstupel an. Hier ist die Antwort für das erste Ergebnistupel „m_1“, da es aus dem Tupel mit der ProvID m_1 stammt, siehe Relation `Messung` in Tabelle 2.2. Die Antwort für das zweite Ergebnistupel ist „m_2“, da es aus dem Tupel mit der ProvID m_2 stammt. Die dritte Teilspalte gibt die Where-Provenance als Kombination von Ursprungsrelation, -tupel und -wert

an. Für das erste Ergebnistupel ist die Antwort „(Messung, m_1, Gründling)“, da das Ergebnis aus dem Attributwert Gründling des Tupels mit der ProvID m_1 aus der Relation `Messung` stammt. Für das zweite Ergebnistupel ist die Antwort „(Messung, m_2, messlab)“, da das Ergebnis aus dem Attributwert messlab des Tupels mit der ProvID m_2 aus der Relation `Messung` stammt.

Why-Provenance Why-Provenance beantwortet die Frage, wieso ein Tupel im Anfrageergebnis enthalten ist. Dazu wird die Menge der sogenannten Zeugen angegeben. Ein Zeuge ist eine Menge von Ursprungstupeln, die beweisen, dass das Tupel Teil des Ergebnisses ist. In Tabelle 2.3 ist die Why-Provenance für beide Ergebnistupel angegeben. Die Why-Provenance des ersten Tupels besteht aus zwei Zeugen, $\{m_1, k_1\}$ und $\{m_1, k_2\}$. Der erste Zeuge $\{m_1, k_1\}$ verweist auf das Tupel mit der ProvID m_1, was beweist, dass die Messung mit der MessungsID 1 den Wert „Gründling“ in der Spalte `Analyst` enthält. Außerdem verweist er auf das Tupel mit der ProvID k_1, welches beweist, dass die Messung mit der MessungsID 1 eine Klasse mit einem Messwert zwischen 0 und 1 enthält. Somit beweist der Zeuge $\{m_1, k_1\}$, dass das erste Tupel der Tabelle 2.3 ein korrektes Ergebnistupel der Anweisung 2.4 ist. Der zweite Zeuge der Zeugenmenge beweist dies ebenso, verweist dabei jedoch auf das Tupel k_2, was eine andere Klasse der MessungsID mit einem Messwert zwischen 0 und 1 ist. Die Why-Provenance des zweiten Tupels des Anfrageergebnisses besteht nur aus einem Zeugen, $\{m_2, k_4\}$, welches nach dem gleichen Prinzip beweist, dass das Tupel korrekterweise Teil des Anfrageergebnisses ist.

How-Provenance How-Provenance beantwortet die Frage, wie das Ergebnis aus den Ursprungstupeln berechnet wurde. Sie wird als Polynom angegeben, wobei Multiplikationszeichen einen Join symbolisieren, und Additionszeichen mehrere Zeugen voneinander trennen. In Tabelle 2.3 ist die How-Provenance für beide Ergebnistupel angegeben. Die How-Provenance des ersten Tupels ist das Polynom $m_1 \cdot k_1 + m_1 \cdot k_2$. Das heißt, dieses Tupel wurde auf zwei verschiedene Arten berechnet: sowohl durch einen Join der Tupel mit den ProvIDs m_1 und k_1, als auch durch einen Join der Tupel mit den ProvIDs m_1 und k_2. Die How-Provenance des zweiten Tupels ist $m_2 \cdot k_4$, d.h. dieses Tupel wurde durch einen Join der Tupel mit den ProvIDs m_2 und k_4 berechnet.

2.5.2 Workflow Provenance

Workflow Provenance legt den Fokus auf die Rückverfolgbarkeit der Arbeitsabläufe von der Erhebung der Primärdaten über eventuell vollzogene Validierungen, Bereinigungen oder andere Änderungen bis hin zum Endprodukt. Es soll dokumentiert werden, wie die Daten entstanden und welche Prozesse sie durchliefen. Ein Arbeitsablauf oder Workflow kann als gerichteter Graph betrachtet werden. Dabei stellen Knoten Funktionen oder Module dar mit Eingaben, Ausgaben und Parametern. Kanten stellen Daten- oder Kontrollflüsse zwischen diesen Modulen dar [HDB17]. Diese Betrachtungsweise erlaubt eine Anpassung der Graphendefinition an verschiedene Anwendungsgebiete und somit Flexibilität, gleichzeitig aber auch eine Grundstruktur, welche das Automatisieren von Workflow Provenance Management erleichtert (siehe Abbildung 2.4.)

2.5.3 Metadata Provenance

Metadata Provenance legt den Fokus auf Informationen über den Entstehungsprozess von Daten, die aus Metadaten gewonnen werden können [HDB17]. Es sind jedoch nicht alle Metadaten für die Metadata Provenance relevant. Zu den relevanten Informationen zählen beispielsweise Zeitstempel der Erstellung und Bearbeitung der Datei, Versionsverläufe, Nutzerkennungen und ähnliches. Mithilfe dieser Informationen lässt sich feststellen, wann die Datei erstellt wurde, von wem und/oder mit welchem Programm

sie erstellt wurde, wann und von wem sie ggf. verändert wurde. Dies ermöglicht die Rekonstruktion des Entstehungsprozesses der Datei.

Wenn beispielsweise das erste Tupel der Relation `Messung` aus Tabelle 2.2 betrachtet wird, könnten u.a. folgende Metadaten zu diesem Tupel vorhanden sein: Name der Relation, Namen und Datentypen der Spalten, Zeitstempel der Erstellung des Tupels und Nutzername des Datenbankanwenders, der dieses Tupel erstellt hat. All diese Informationen sind Metadaten, jedoch nur der Zeitstempel und Nutzername betreffen den Entstehungsprozess dieses Tupels.

Bei Metadata Provenance handelt es sich um die allgemeinste Form der Provenance. Die verfügbaren Informationen können von Datei zu Datei stark unterschiedlich sein. Somit lassen sich keine allgemein gültigen Restriktionen, Annahmen oder Eigenschaften für Metadata Provenance definieren. Dementsprechend ist auch die Automatisierung von Metadata Provenance Management schwer umzusetzen (siehe Abbildung 2.4.)

Im folgenden Kapitel werden der Stand der Forschung und der Stand der Technik beschrieben. Es werden andere Arbeiten vorgestellt, die sich mit verwandten Themen beschäftigen und das Datenbankmanagementsystem MySQL vorgestellt.

3 State of the Art

In diesem Kapitel wird der aktuelle Stand der Forschung und der Technik beschrieben. Zunächst werden in Abschnitt 3.1 einige wissenschaftliche Arbeiten vorgestellt, die sich mit verwandten Themen beschäftigen. Danach erfolgt in Abschnitt 3.2 ein Überblick über das Datenbankmanagementsystem MySQL, welches für die Implementierung der Datenbank genutzt wird.

3.1 Stand der Forschung

In diesem Abschnitt werden einige Arbeiten vorgestellt, die sich mit ähnlichen Aufgabenstellungen beschäftigen. Da zwischen dem Leibniz-Institut für Ostseeforschung Warnemünde und der Universität Rostock eine enge Zusammenarbeit herrscht, sind bereits einige Artikel und studentische Arbeiten zum Forschungsdatenmanagement am IOW entstanden.

Der Artikel [BKM+17] gibt eine Übersicht über das Forschungsdatenmanagement am IOW. Er beschreibt den allgemeinen Workflow von Forschungsdaten anhand von Beispielen aus Projekten des IOW und zeigt verschiedene Forschungsfragen in den Bereichen Datenintegration, temporale Datenbanken und Data Provenance auf. Es werden einige Herausforderungen des Datenmanagements thematisiert, die auch in dieser Bachelorarbeit relevant sind. Dabei handelt es sich unter Anderem um die Ausgangslage vorhandener Daten in Form von dezentraler Datenhaltung und heterogenen Speicherformaten, die die Erstellung eines einheitlichen Datenmodells aufwändig macht, das Vorhandensein von Medienbrüchen bei der Entstehung der Daten, sowie die Veränderungen von Technik und Messmethoden im Laufe der Zeit.

In der Bachelorarbeit [Man20] wird untersucht, wie sich Schema-Änderungen in einer Datenbank auf die Anwendbarkeit von Provenance-Techniken auswirken. Als Datengrundlage der Arbeit dienen CTD-Daten[1] vom IOW, deren Speicherstrukturen sich im Laufe der Jahrzehnte stark änderten. Es werden mehrere Datenbankschemata für unterschiedliche Zeitabschnitte der Daten erstellt, und die Unterschiede zwischen den Schemata mithilfe von Schema-Evolutions-Operatoren beschrieben. Die Schema-Änderungen werden auf Kompatibilität mit Data Provenance untersucht. Der Autor schlägt eine neue Art von Data Provenance vor, die sogenannte *What-Provenance*, welche nach Abschluss der Bachelorarbeit in [AMJ+20] formal definiert wurde. Wird What-Provenance auf eine Datenbankanfrage angewendet, werden die Datentypen der Attribute der Ursprungstupel ausgegeben. Mithilfe der neuen What-Provenance und bisher bekannten Arten der Data Provenance kommt der Autor zu dem Schluss, dass die meisten Schema-Evolutions-Operatoren mit Data Provenance kompatibel sind.

Die Bachelorarbeit [Mö16] beschäftigt sich mit der Forschungsdatenverwaltung des GODESS-Projekts aus der Sektion Meereschemie am IOW. Dabei handelt es sich um eine Messstation, die durch verschiedene Messinstrumente und Sensoren automatisch eine Vielzahl von chemischen und physikalischen Parametern der Ostsee misst. Diese Daten werden, wie die Sedimentdaten dieser Arbeit, dezentral gespeichert und können nur schwer dateiübergreifend analysiert werden. In der Arbeit wurde unter Berücksichtigung von

[1] CTD steht für Conductivity, Temperature und Depth, also die elektrische Leitfähigkeit, Temperatur und Tiefe des Wassers

Provenance-Anforderungen ein prototypisches Framework zur Speicherung der GODESS-Daten entworfen. Dabei wurde vor allem eine Kombination von temporalen Aspekten und Metadaten eingesetzt, um Provenance-Techniken umzusetzen. Dieser Ansatz wurde in der vorliegenden Bachelorarbeit nicht weiter verfolgt, da hier der Fokus auf Data Provenance gelegt wird und weitaus weniger Metadaten vorhanden sind als bei den GODESS-Daten.

Schema-Änderungen in einer Datenbank und neue Analyseverfahren führen in der Regel zu Änderungen von Anfragen auf die Daten. In der Masterarbeit [Mey16] wird ein Versionierungskonzept für Änderungen von SQL-Funktionen entwickelt. Dieses soll in Verbindung mit Versionierung der Schemaänderungen die Reproduzierbarkeit von Anfragen und Analysen zu beliebigen Zeiträumen ermöglichen, und die Anwendung neuer Funktionen auf alte Datenbestände erleichtern. Auch hier wurde als Lösungsansatz der Fokus auf temporale Datenhaltung gesetzt.

Diese Arbeiten wurden aufgrund des zentralen Themas Forschungsdatenmanagement am IOW ausgewählt. Sie lieferten verschiedene Ansätze für den Datenbankentwurf und die Umsetzung verschiedener Provenance-Techniken.

3.2 Stand der Technik

In dieser Arbeit wird das Datenbankmanagementsystem MySQL verwendet. Es handelt sich dabei um eines der verbreitetsten relationalen Datenbankmanagementsysteme und ist sowohl als kostenlose Open Source Software, als auch als kommerzielle Software verfügbar. Es wurde sich für die Verwendung von MySQL entschieden, da es auch am IOW eingesetzt wird und somit die Kompabilität der hier entworfenen Datenbank mit dem System des IOW gewährleistet werden kann. Konkret wird in dieser Arbeit die Open Source Variante MySQL Community Edition und die graphische Benutzeroberfläche MySQL Workbench in Version 8.0.28 genutzt. Alle SQL-Statements zum Erstellen und Modifizieren der Datenbank wurden in der MySQL Workbench ausgeführt. Zur Datenintegration wurde primär der Table Data Import Wizard der MySQL Workbench genutzt, ein Teil der Datenintegration wurde jedoch manuell durch SQL-Statements vorgenommen.

Für die in dieser Arbeit angewendeten Techniken und Verfahren ist MySQL eine geeignete Wahl und entspricht den Anforderungen. Sollte der hier erstellte Datenbankentwurf jedoch in Zukunft noch um andere Provenancetechniken, temporale Aspekte oder komplexe Datenauswertungen erweitert werden, könnte der Funktionsumfang von MySQL nicht länger ausreichend sein, wie in [Mö16] festgestellt wurde.

4 Eigene Konzeptentwicklung

Dieses Kapitel beschreibt das Konzept zur Lösung der Problemstellung. Zuerst werden die genauen Daten vorgestellt, auf denen diese Arbeit basiert. Dabei wird für jede Datenstruktur ein eigenes ER-Modell erstellt. Diese Modelle werden zu einem Gesamtmodell kombiniert. Das gesamte ER-Modell wird um Provenancetechniken erweitert und auf ein Relationenmodell abgebildet, auf dessen Basis eine Datenbank erstellt werden kann. Zuletzt wird erläutert, wie die Daten vorbereitet werden müssen, um in die Datenbank importiert werden zu können.

4.1 Vorliegende Daten

Bei den vorliegenden Daten handelt es sich um Korngrößenanalysen von Sedimentproben aus der Ostsee, also kleine Proben des Meeresbodens, der z.B. aus Sand, Kies oder Schlick besteht. Die Daten sind unterschiedlichen Ursprungs. Ein Teil der Daten, die sogenannten „Altdaten“ wurden von unterschiedlichen Einrichtungen erhoben und durch das IOW vor einigen Jahren zusammen gestellt. . Die ältesten zur Verfügung stehenden sedimentologischen Daten des IOW stammen von ca. 1950. Diese Datensammlung wurde 2011 für sich abgeschlossen, d.h. es kommen in diesem Format keine weiteren Daten mehr hinzu. Daten neueren Datums liegen in individuellen Strukturen und Dateiformaten vor.

Die Proben werden auf Forschungsfahrten mit Greifern vom Meeresboden entnommen und oft noch an Bord des Schiffs makroskopisch beschrieben, wobei die Beschreibungen meist handschriftlich aufgezeichnet und später in eine Excel-Tabelle übertragen werden. Nach Ende der Fahrt werden im Labor Korngrößenmessungen durchgeführt, d.h. es wird untersucht, wie hoch der Anteil von Sedimentkörnern verschiedener Durchmesser ist. Dabei gibt es verschiedene Methoden zur Aufbereitung der Proben sowie zur Messung und Klassifizierung der Korngrößen. Aktuell werden im IOW zur Messung der Sedimentproben zwei verschiedene Messmethoden verwendet.

Zum Einen können die Proben gesiebt werden, wobei die Anzahl der genutzten Siebe der Anzahl von Klassen entspricht und die Maschengrößen der Siebe die Klassengrenzen bestimmt. Wie in Abbildung 4.1a zu sehen, werden hierfür mehrere Siebe übereinander gestapelt, wobei die Maschengröße nach unten abnimmt. Die Sedimentprobe wird manuell gewogen und dann in das oberste Sieb gegeben. Der gesamte Stapel wird für mehrere Minuten auf einem Rütteltisch durchgesiebt und danach wird der Inhalt der einzelnen Siebe gewogen. Da die Messungen überwiegend manuell durchgeführt und dokumentiert werden, sind die Ergebnisse teilweise in unterschiedlichen Schemata und Formaten gespeichert. Dies sorgt für eine hohe Diversität der gespeicherten Informationen und erhöht den Aufwand, der mit der Anpassung der Daten zum Import in eine Datenbank verbunden ist.

Die andere verwendete Methode nutzt ein Laserbeugungs-Partikelmessgerät, den Mastersizer 3000, zu sehen in Abbildung 4.1b. Dieses Gerät kann durch verschiedenes Zubehör sowohl trockene als auch nasse Sedimentproben verarbeiten. Das Gerät ermittelt die Korngrößen, indem es die Sedimentpartikel mit einem Laser bestrahlt und die Beugung des Laserstrahls nach Auftreffen auf die Partikel misst. Da kleine Partikel das Licht in einem größeren Winkel beugen als große Partikel, kann der Mastersizer so die

(a) Stapel von vier Sieben, Bildquelle: Dr. Svenja Papenmeier

(b) Mastersizer 3000 mit Zubehör für nasse und trockene Proben, Bildquelle: Malvern Panalytical[1]

Abbildung 4.1: Messmethoden

Größe der Partikel aus den Beugungswinkeln berechnen. Um natürliche Messabweichungen zu erkennen, wird eine Probe meist mehrfach hintereinander untersucht und der Durchschnitt aus den Messergebnissen berechnet. Die einzelnen Messungen und die Durchschnittswerte werden gemeinsam in einer Datei gespeichert. Die Ergebnisse können dann von der Software des Mastersizers direkt in verschiedene Formate exportiert werden. Hier gibt es ein vom Institut vorgeschriebenes Schema und Format, in dem alle Mastersizer-Daten gespeichert werden sollen. Die nass und trocken gemessenen Proben werden mit dem gleichen Schema und Format exportiert. Sie werden allerdings in unterschiedliche Dateien exportiert, um die Messmethodik nachvollziehbar zu machen, da diese Information nicht explizit in den Exporten enthalten ist. Da die Mastersizer-Dateien alle sehr ähnlich strukturiert sind, lässt sich der Import dieser Daten in eine Datenbank einfacher automatisieren als für Siebdaten.

Zusätzlich zu den bereits erwähnten Altdaten wurden für diese Arbeit vom IOW exemplarische Datensätze aller Messmethoden zur Verfügung gestellt. Diese „Neudaten“ umfassen die Datensätze AT2020, Atlas Projekt und Mastersizer. Für einige Datensätze liegen in separaten Tabellen Metadaten vor. In den folgenden Abschnitten werden die einzelnen Datenstrukturen genauer vorgestellt. Die Namen der Abschnitte basieren auf den Dateiordnern, in denen die jeweiligen Daten zur Verfügung gestellt wurden. In jedem Abschnitt wird eine Datenstruktur analysiert und ein ER-Modell dazu erstellt. Die Namen der Attribute werden hierbei möglichst exakt aus den Daten entnommen. Daher sind die Namen der Elemente in den ER-Modellen in manchen Abschnitten deutsch und in anderen Abschnitten englisch.

4.1.1 Altdaten

Im Ordner Altdaten befindet sich eine Zusammenstellung von Sedimentdaten aus verschiedenen Quellen. Die Daten enthalten keine Informationen zu den verwendeten Messmethoden. Aufgrund der Anzahl an Datensätzen, der sehr unterschiedlichen Anzahl von gemessenen Korngrößenklassen je Datensatz ist jedoch anzunehmen, dass verschiedene Messmethoden wie Sieb- oder Mastersizer-Messungen enthalten sind. Die ca. 35 000 Datensätze sind tabellarisch in der Datei `all_cum.txt` gespeichert, zu sehen in Abbildung 4.2. Die Datei info.txt enthält Erklärungen einiger Spaltennamen und eine Legende der Abkürzungen in der Spalte quelle. Es sind folgende Spalten vorhanden:

[1] `https://www.malvernpanalytical.com/de/support/product-support/mastersizer-range/mastersizer-3000`

- idx_map: Diese Spalte wird nicht in `info.txt` erklärt. Es könnte sich dabei um eine Art von Proben-ID handeln. Es gibt jedoch zahlreiche mehrfach vorkommende Werte und negative Werte in der Spalte. Daher ist sie nicht als Schlüssel geeignet.

- stat: Die Spalte stat wird auch nicht in `info.txt` erklärt. Vermutlich handelt es sich um eine Stations-ID. Die Werte sind je nach Quelle der Zeile rein numerisch oder alphanumerisch.

- lon_wgs84 und lat_wgs84: Auch diese Spalten werden nicht in `info.txt` erklärt. Es handelt sich dabei um die Koordinaten, an denen die Probe genommen wurde.

- quelle: Die Spalte quelle enthält eine Abkürzung der Quelle, aus der der Datensatz stammt. Dabei gibt es verschiedene Kürzel für Datensätze des IOW aus verschiedenen Zeitabschnitten, sowie für andere Institute und Firmen. Viele der enthaltenen Kürzel sind in `info.txt` erklärt, es gibt jedoch auch Kürzel in der Tabelle, die nicht in `info.txt` enthalten sind.

- n: Die Spalte n zeigt an, in wie viele Korngrößenklassen die Probe unterteilt wurde. Dementsprechend enthält der Datensatz n Paare von Klassengrenze und Wert. Der Wertebereich von n liegt zwischen 1 und 161.

- d*: Die Spalten d01 bis dn enthalten die Grenzen der Korngrößenklassen der Probe. Die Werte sind als Durchmesser in μm angegeben.

- cum*: Die Spalten cum01 bis cumn enthalten die dazugehörigen *kumulativen* Messwerte der Korngrößenklassen mit einem Wertebereich von 0 bis 1. Der kumulative Messwert einer Korngrößenklasse ist der Anteil an Körnern der gesamten Sedimentprobe, der größer als der Grenzwert der Klasse ist.

	A	B	C	D	E	F	G	H	I	J	K	L	M	N
1	idx_map	stat	lon_wgs84	lat_wgs84	quelle	n	d01	cum01	d02	cum02	d03	cum03	d04	cum04
2														
3	10266	7150	12.44752	54.69833	iow_a	3	200	0.0166	100	0.7957	60	0.9701		
4														
5	10268	7170	12.44752	54.66499	iow_a	3	200	0.0201	100	0.944	60	0.9893		
6														
7	10269	7180	12.44752	54.64833	iow_a	3	200	0.0234	100	0.9522	60	0.9954		
8														
9	10270	7190	12.47668	54.64833	iow_a	3	200	0.0052	100	0.9375	60	0.9909		
10														
11	10271	7200	12.47668	54.665	iow_a	3	200	0.0046	100	0.8884	60	0.9718		
12														
13	10272	7210	12.47668	54.68166	iow_a	3	200	0.0193	100	0.8623	60	0.9799		
14														
15	10273	7220	12.47668	54.69833	iow_a	3	200	0.0067	100	0.7742	60	0.9663		

Abbildung 4.2: Ausschnitt aus `all_cum.txt`

Basierend auf diesen Informationen kann nun ein ER-Modell für die Altdaten erstellt werden, zu sehen in Abbildung 4.3. Aufgrund des großen Wertebereichs von n ist die Länge der Datensätze sehr unterschiedlich. Daher wurden die Daten auf zwei Entity-Typen aufgeteilt: Probe und Klasse. Probe enthält die Informationen aus den Spalten `idx_map`, `stat`, `quelle` und die Koordinaten aus `lon_wgs84` und `lat_wgs84` als Attribute. Zusätzlich wurde ein künstlicher Schlüssel `ProbenID` eingefügt, da die Daten keine als Schlüssel geeignete Attributmenge enthalten. Der Entity-Typ Klasse enthält die Wertepaare, wobei das Attribut `Größe` die Informationen aus d* und das Attribut `Wert` die Informationen aus cum* enthält. Klasse ist funktional abhängig von Probe, d.h. Instanzen von Klasse können nicht existieren, ohne einer Instanz von Probe zugeordnet zu sein. Jede Instanz von Klasse ist exakt einer Instanz von Probe zugeordnet. Aber einer Instanz von Probe können mehrere Instanzen von Klasse zugeordnet sein. Die Anzahl von Klassen, die einer Probe zugeordnet sind, entspricht dem Wert der Spalte n dieser Probe. Daher ist diese bereits implizit im ER-Modell vorhanden und muss nicht als Attribut eingefügt werden.

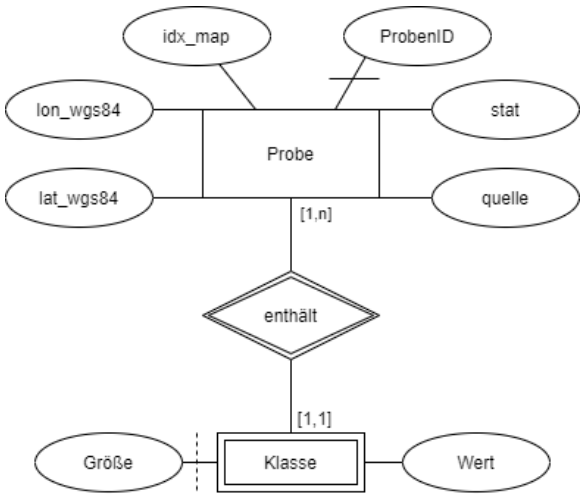

Abbildung 4.3: ER-Modell für die Altdaten

4.1.2 AT2020

Der Ordner AT2020 enthält die Datei `AT2020_Grainsize.xlsx`, in welcher Siebergebnisse von Proben tabellarisch dokumentiert wurden. In der Datei befinden sich mehrere Reiter. Für diese Arbeit ist der Reiter „Sieben - Split eingerechnet“ relevant. Dieser Reiter enthält die Siebergebnisse in zwei Einheiten, der obere Teil der Tabelle zeigt die Originaldaten in Gramm und der untere Teil zeigt die oberen Daten konvertiert in prozentuale Anteile an der Gesamtmenge. Außerdem ist die Tabelle vertikal unterteilt in Details zur Messung wie Messmethode, Gesamtgewicht der Probe und Probennummer auf der linken Seite (siehe Abbildung 4.4a) und die Messergebnisse mit Klassengrößen und Werten auf der rechten Seite (siehe Abbildung 4.4b).

Die linke Seite der Tabelle enthält folgende Spalten:

- Bemerkung: Platz für zusätzliche Bemerkungen und Kommentare
- gesplittet bei 2mm: In dieser Spalte wird eingetragen, ob die Probe bei 2mm Siebdurchmesser gesplittet wurde. Eine zu große Menge an Probenmaterial kann die Siebe verstopfen und somit Messergebnisse verfälschen. Daher werden große Proben ggf. gesplittet und nur ein Teil der Probe untersucht.
- Messmethode: In dieser Spalte wird eingetragen, mit welcher Methode die Probe untersucht wurde.
- $< 63\mu m$ abgeschlämmt: In dieser Spalte wird eingetragen, ob die Probe unter $63\mu m$ Siebdurchmesser abgeschlämmt wurde. Ab einer Sedimentgröße von ca. $63\mu m$ ist ein weiteres Trockensieben kaum möglich, da die kleinen Partikel durch elektrostatische Aufladung am Sieb hängen bleiben. Daher wird die Probe bei $63\mu m$ gewogen und dann im Sieb mit Wasser gespült, bis die Partikel $< 63\mu m$ entfernt wurden. Der Rest wird getrocknet und erneut gewogen. So lässt sich die Menge an Sedimenten $< 63\mu m$ aus der Gewichtsdifferenz berechnen.
- gesamt Gewicht trocken: In dieser Spalte wird das Gesamtgewicht der trockenen Probe eingetragen.
- trocken nach Schlämmen: Falls die Probe abgeschlämmt wurde, wird in dieser Spalte das Gewicht der getrockneten Probe nach dem Schlämmen eingetragen.
- Summe aller Klassen: In dieser Spalte wird die Summe der Messergebnisse aller Klassen eingetragen, wodurch der Siebverlust berechnet werden kann.

	A	B	C	D	E	F	G	H	I	J	K	L
1												
2	Gramm											
3												Siebnr
4												
5		Bemerkung	gesplittet bei 2 mm	Messmethode	<63 µm abgesc	gesamt Gewich	trocken nach Schlä	Summe aller Kl	Siebverlust	%	Muschelschalen	Probe
6				Sieben	nein	465.75		463.95	-1.8	-0.4		AT2020_01
7			ja	Sieben	nein	405.94		395.26	-10.68	-2.6		AT2020_02
8			ja	Sieben	nein	477.88		475.77	-2.11	-0.4		AT2020_03
9				Sieben	nein	500.72		510.69	9.97	2.0		AT2020_05
10			ja	Sieben	nein	554.97		529.89	-25.08	-4.5		AT2020_06
11			ja	Sieben	nein	770.93		774.68	3.75	0.5		AT2020_07
12			ja	Sieben	nein	504.48		511.91	7.43	1.5		AT2020_09
13				Sieben	ja	776.01	731.85	773.13	-1.42	-0.2	1.46	AT2020_10
14			ja	Sieben	ja	678.59	651.65	682.36	3.77	0.6		AT2020_11
15				Sieben	ja	96.42	72.93	96.97	0.55	0.6		AT2020_15
16				Sieben	ja	307.82	279.52	307.61	-0.21	-0.1		AT2020_19
17				Sieben	ja	131.45	116.44	132.17	0.72	0.5		AT2020_23
18				Sieben	nein	403.46	358.58	402.85	-0.61	-0.2		AT2020_24
19				Sieben	ja	138.97	130.75	131.02	-2.92	-2.1	5.03	AT2020_26
20		2. mal gesiebt von Svenja		Sieben	nein	476.82		475.49	-1.33	-0.3		AT2020_03
21		2. mal gesiebt von Svenja		Sieben	nein	560.2		557.72	-2.48	-0.4		AT2020_06
22												
23												
24	%											
25				Sieben	nein	465.75		463.95	-1.8	-0.4		AT2020_01_S
26			ja	Sieben	nein	405.94		395.26	-10.68	-2.6		AT2020_02_S
27			ja	Sieben	nein	477.88		502.436666667	24.55666667	5.1		AT2020_03_S

(a) linke Seite mit Details zur Messung

	A	L	N	O	P	Q	R	S	T	U	V	W	X	Y	Z
1															
2	Gramm		medium gravel and larger						fine gravel						very fine gra
3		Siebnr	2	3	4	5	6	7	8	9	10	11	12	13	14
4			16	13.2	12.5	11.2	9.5	8	6.7	6.3	5.6	4.75	4	3.35	2.8
5		Probe	16000	13200	12500	11200	9500	8000	6700	6300	5600	4750	4000	3350	2800
6		AT2020_01	26.21	23.46	3.09	9.04	9.35	13.75	13.41	2.15	5.06	5.83	11.72	13.48	19.18
7		AT2020_02	9.48	6.99	0	5.18	4.82	1	6.12	2.2	2.8	5.21	5.65	5.42	5.75
8		AT2020_03	0	0	0	0	0	0	1.23	0	2.42	2.72	5.74	4.89	12.79
9		AT2020_05	0	0	3.08	0	3.11	2.22	2.77	2.48	2.3	4.24	4.13	3.91	6.68
10		AT2020_06	10.21	16.12	0	17.39	3.77	17.94	14.36	4.29	7.74	11.37	13.75	17.22	23.52
11		AT2020_07	28.93	4.59	3.24	14.41	18.47	13.7	6.39	5.52	7.4	11.6	14.02	11.21	12.75
12		AT2020_09	9.79	15.27	0	0	11.14	6.35	8.01	4.42	6.47	9.69	9.4	10.81	13.75
13		AT2020_10	57.83	10.42	3.15	3.85	4.46	8.2	5.86	3.79	4.64	7.29	11.31	8.81	9.73
14		AT2020_11	15.83	0	0	2.38	3.09	5.75	5.81	2.62	2.71	3.43	6.05	5.43	8.06
15		AT2020_15	12.88	0	5.77	5.34	6.2	4.94	5.78	1.74	1.49	1.66	2.2	1.36	1.42
16		AT2020_19	25.3	4.52	0	0	1.47	1.09	0	0.05	0.22	0.04	0.5	0.79	0.66
17		AT2020_23	0	8.65	3.68	0	1.99	0	0.72	0	0.38	0.4	0.84	0.32	0.47
18		AT2020_24	0	19.12	0	3.91	2.57	3.46	0.64	0.23	1.48	2.27	1.18	1.32	1.79
19		AT2020_26	0	0	0	0	0	1.92	0	0.82	0	0.86	0.51	0.99	1.06
20		AT2020_03	0	0	0	0	0	0	1.7	0	2.2	3.5	4.79	5.6	13.63
21		AT2020_06	10.21	0	16.1	15.53	6.66	16.61	13.76	4.63	7.7	10.85	13.88	17.36	23.26
22															
23															
24	%		16000	13200	12500	11200	9500	8000	6700	6300	5600	4750	4000	3350	2800
25		AT2020_01_S	5.64931566	5.05657937	0.66602005	1.94848583	2.01530337	2.96368143	2.890397672	0.46341201	1.09063477	1.25660093	2.52613428	2.905485505	4.13406617
26		AT2020_02_S	2.39842129	1.76845621	0	1.31052978	1.21945049	0.25299803	1.548347923	0.55659566	0.70839447	1.31811972	1.42943885	1.371249304	1.45473865
27		AT2020_03_S	0	0	0	0	0	0	0.244806974	0	0.48165275	0.54136176	1.14243255	0.973256994	2.54559447

(b) rechte Seite mit Messergebnissen

Abbildung 4.4: Ausschnitte aus der Datei `AT2020_Grainsize.xlsx`

- Siebverlust: In dieser Spalte wird der Siebverlust eingetragen, d.h. die Differenz zwischen dem trockenen Gesamtgewicht und der Summe aller Klassen.
- % : In dieser Spalte wird der Siebverlust als prozentualer Anteil vom Gesamtgewicht eingetragen.
- Muschelschalen: In dieser Spalte wird eingetragen, wie viel Gramm Muschelschalen in der Probe enthalten waren.
- Probe: In dieser Spalte wird die Probennummer eingetragen.

Auf der rechten Seite der Tabelle sind die Messergebnisse eingetragen, wobei die Klassengrenzen mehrmals in verschiedenen Formen angegeben sind. In der zweiten Zeile des Headers werden mehrere Klassen in eine Oberkategorie zusammengefasst, z.B. fine gravel, coarse sand, fine sand usw. In der dritten Zeile enthält die Nummer des verwendeten Siebes. In der vierten Zeile ist die Klassengröße in mm angegeben und in der fünften Zeile in μm.

Basierend auf diesen Informationen kann nun ein ER-Modell für die AT2020-Daten erstellt werden, siehe Abbildung 4.5. Die Daten werden wieder in zwei Entity-Typen Probe und Klasse unterteilt, wobei Klasse funktional abhängig von Probe ist. Die Spalte Probe aus der Datei `AT2020_Grainsize.xlsx`

wird als das Attribut `Probennummer` übernommen, aber da mehrere Messungen von der gleichen Probe in einer Datei gespeichert werden können, wird zusätzlich das Attribut `ProbenID` als künstlicher Schlüssel eingeführt. Die Spalten $< 63\mu$`m abgeschlämmt`, `trocken nach Schlämmung`, `Gesamtgewicht`, `Muschelschalen`, `gesplittet bei 2mm`, `Bemerkung`, `Siebverlust`, `Summe aller Klassen` und `Messmethode` werden als Attribute von Probe übernommen. Die Messwerte und Korngrößen aus der rechten Seite der Tabelle werden in den Attributen `Größe` und `Wert` von Klasse übernommen. Die Oberkategorien und Siebnummern der rechten Seite werden nicht in das ER-Modell aufgenommen, da die Oberkategorie durch die gespeicherte Korngröße ermittelt werden kann und die Siebnummer keine aussagekräftige Information beinhaltet. Die Siebnummer stellt lediglich die Reihenfolge der genutzten Siebe in diesem Messvorgang dar.

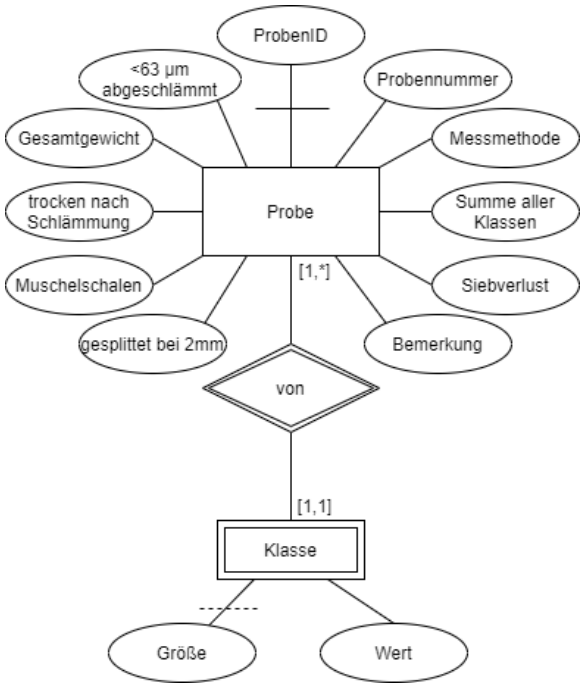

Abbildung 4.5: ER-Modell für AT2020

4.1.3 Atlas Projekt

Im Ordner Atlas Projekt ist die Datei `Alle_Sedimentproben_PG.xlsx` von Relevanz für diese Arbeit. Die Datei ist tabellarisch aufgebaut, im Gegensatz zu den anderen Dateistrukturen sind hier jedoch Zeilen und Spalten umgekehrt, d.h. eine Spalte enthält einen Datensatz und eine Zeile entspricht einen Parameter. Die Datei enthält zwei Reiter, „gesiebt_bio“ und „Mastersizer“. Die Daten im Mastersizer-Reiter sind, bis auf die umgekehrten Achsen, den Daten im Kapitel 4.1.4 sehr ähnlich und werden deshalb in diesem Kapitel nicht näher beschrieben. In Abbildung 4.6 sind die Siebdaten im Reiter „gesiebt_bio“ zu sehen.

Die Tabelle enthält folgende Zeilen:

- Sample Identity: Die Zeile Sample Identity enthält die Probennummern der Proben.
- Analyst: Die Zeile Analyst enthält den Namen der Person, welche die Messung durchgeführt hat.
- Date: Die Zeile Date enthält den Zeitpunkt der Messung.
- Initial Sample Weight: Die Zeile Initial Sample Weight enthält das Gesamtgewicht der Probe.

	A	B	C	D	E	F	G	H	I	J	K	L	M
1	Sample Identity:	ATLAS_PG34	ATLAS_PG36	ATLAS_PG37	ATLAS_PG38	ATLAS_PG39	ATLAS_PG40	ATLAS_PG43	ATLAS_PG44	ATLAS_PG45	ATLAS_PG46	ATLAS_PG47	ATLAS_PG4
2	Analyst:												
3	Date:												
4	Initial Sample Weight:	100	100	100	100	100	100	100	100	100	100	100	100
5	0	0.2	0.1	1.6	0.1	0	0.2	0	0.1	0.1	1.3	0.3	0.2
6	63	1	0.4	8.9	0.2	0.1	1.3	0.4	0.2	0.9	3.1	2.2	0.9
7	90	7.6	4.1	51.7	1.9	0.6	16.2	3.7	0.3	9.2	17.3	19.7	2.2
8	125	25.2	15.9	31.3	11.8	7.1	46.9	24.7	0.7	16.6	26.6	30.5	5.2
9	180	48.2	50.7	4.6	17.3	28.5	27.7	24.7	4.8	44.9	17.6	25.8	24.4
10	250	16.1	25.2	1.1	23.8	32.7	5.6	19.7	30.5	22.4	18.1	15.1	37.4
11	355	1.5	2.8	0.5	23.2	21.3	0.6	11.4	19.9	4.3	7.8	4.5	17.1
12	500	0.2	0.5	0.1	12.3	6.1	0.2	6.5	14.5	1.2	2.9	1.3	6.4
13	710	0	0.2	0.1	4.6	2	0.2	4.1	20.7	0.3	1.2	0.4	2.9
14	1000	0	0.1	0	1.7	0.8	0.1	2.4	6	0.1	0.8	0.1	1.2
15	1400	0	0.1	0	1.7	0.5	0.2	1.8	1.5	0.1	0.8	0.1	0.8
16	2000	0	0	0	1.4	0.3	0.7	0.5	0.4	0.1	0.3	0.1	0.1
17	2360								0		0.4		0.2
18	2800								0.1		0.6		0.3
19	3350								0.2		0.5		0.4
20	4000								0.2		0.8		0.2

Abbildung 4.6: Ausschnitt aus der Datei `Alle_Sedimentproben_PG.xlsx`

Danach folgen die verschiedenen Klassengrößen in μm und die dazugehörigen Messwerte. Im Datenausschnitt in Abbildung 4.6 sind die Messwerte in Prozent angegeben, daher sind alle Werte in der Zeile Initial Sample Weight 100. Außerdem wurden in diesem Ausschnitt die Zeilen Analyst und Date nicht ausgefüllt.

Basierend auf diesen Informationen kann nun ein ER-Modell für die Atlas-Projekt-Daten erstellt werden, welches in Abbildung 4.7 zu sehen ist. Auch in diesem Modell wurden die Daten wieder in zwei Entity-Typen aufgeteilt, Sample und Class. Class ist funktional abhängig von Sample und enthält die Klassengrößen im Attribut `Size` und die Messwerte im Attribut `Value`. Die Zeilen `Sample Weight`, `Sample Identity`, `Analyst` und `Date` wurden als Attribute von Sample übernommen, wobei das Attribut `Sample Identity` den Schlüssel bildet.

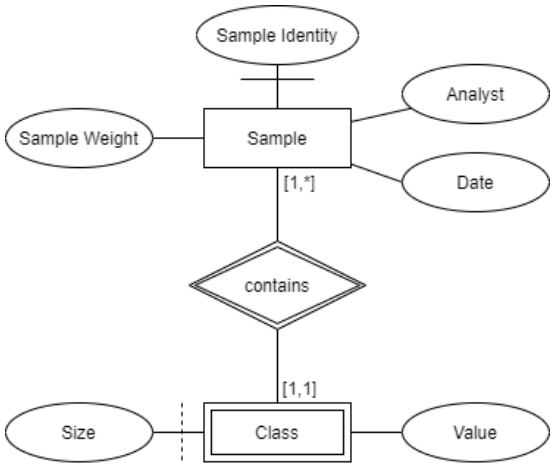

Abbildung 4.7: ER-Modell für das Atlas Projekt

4.1.4 Mastersizer

Bei den Daten im Ordner Mastersizer handelt es sich nicht um Siebdaten, sondern um Daten von Messungen, die mittels eines Laserbeugungs-Partikelmessgerät namens Mastersizer 3000 durchgeführt wurden. Wie in Kapitel 4.1 beschrieben, kann der Mastersizer sowohl trockene als auch nasse Sedimentproben verarbeiten. Diese werden in separaten Dateien mit der gleichen Struktur gespeichert, im Ordner Mastersizer liegen also die Dateien `AT2020_Aero.txt` für trockene Proben und `AT2020_wet.txt` für nasse Proben (siehe Abbildung 4.8).

Die Dateien enthalten folgende Spalten:

	A	B	C	D	E	F	G	H	I
1	Record Number	Sample Name	Measurement Date Time	Dx (10)	Dx (50)	Dx (90)	Operator Name	0.01	0.01136163100452
2	1	AT2020_18	08/13/20 09:48 AM	46.3900529278698	152.652919226921	257.630352005439	messlab	0	0
3	2	AT2020_18	08/13/20 09:48 AM	44.6955984038162	152.804339535208	258.063946561299	messlab	0	0
4	3	AT2020_18	08/13/20 09:49 AM	41.5504537590232	151.522720535801	255.97256792609	messlab	0	0
5	4	AT2020_18	08/13/20 09:49 AM	40.5676305196525	151.118233904254	255.76840406961	messlab	0	0
6	5	AT2020_18	08/13/20 09:49 AM	45.4801318990478	154.29066458417	258.586868213713	messlab	0	0
7	6	Average of 'AT2020_18'	08/13/20 09:48 AM	43.6493173413749	152.477582863685	257.220211761155	messlab	0	0
8	7	AT2020_23_1	08/13/20 10:09 AM	14.4414409229474	115.039138880199	567.391667630885	messlab	0	0
9	8	AT2020_23_1	08/13/20 10:10 AM	15.1545781164109	120.54088477478	593.537333591778	messlab	0	0
10	9	AT2020_23_1	08/13/20 10:10 AM	13.8856914346923	114.085230885729	533.154630198102	messlab	0	0
11	10	AT2020_23_1	08/13/20 10:10 AM	13.7890908412186	113.835449844516	531.805411218671	messlab	0	0
12	11	AT2020_23_1	08/13/20 10:11 AM	14.5866516271991	118.470934102527	545.937493744396	messlab	0	0
13	12	Average of 'AT2020_23_1'	08/13/20 10:09 AM	14.3567894626993	116.322219544842	553.317496889746	messlab	0	0

Abbildung 4.8: Auszug aus der Datei `AT2020_wet.txt`

- Record Number: Die Spalte Record Number enthält eine je Datei fortlaufende Nummer, welche angibt, um die wievielte Messung es sich in dieser Datei handelt. Durchschnittswerte werden dabei auch als Messung gezählt.
- Sample Name: Die Spalte Sample Name enthält die Probennummer des Datensatzes.
- Measurement Date Time: Die Spalte Measurement Date Time enthält den Zeitstempel der Messung.
- Dx(n): Die Spalten Dx(n) enthalten einen statistischen Wert, der angibt, ab welcher Korngröße n% der Probe enthalten sind. Der Wert 46,39 in Zelle D2 in Abbildung 4.8 bedeutet also, dass in dieser Messung 10% der Sedimentkörner kleiner als 46,39 μm sind. Der Wert kann für jedes n zwischen 0 und 100 berechnet werden und manche Mastersizer-Dateien enthalten mehr Dx(n)-Spalten mit verschiedenen n-Werten, jedoch sind im vom Institut vorgeschriebenen Schema nur die Werte 10, 50 und 90 vorgesehen.
- Operator Name: Die Spalte Operator Name enthält den Namen der Person, die die Messung durchgeführt hat. Wenn jedoch kein Name explizit eingegeben wurde, wird automatisch messlab als Standardwert eingetragen.

In den Spalten danach folgen die Klassengrößen in μm und die dazugehörigen Messwerte in Prozent. Um natürliche Messabweichungen zu ermitteln, werden die Proben in der Regel mehrfach hintereinander gemessen und der Durchschnitt der Messungen berechnet und als zusätzliche Zeile in die Datei eingetragen.

Basierend auf diesen Informationen kann nun ein ER-Modell für die Mastersizer-Daten erstellt werden (siehe Abbildung 4.9). Wie in den bisherigen Unterkapiteln werden die Messergebnisse in einen eigenen Entity-Typ Class mit den Attributen `Value` und `Size` ausgelagert. Da die Proben sehr oft mehrfach gemessen werden, wird zusätzlich eine Unterteilung der Daten in die Entity-Typen Sample und Measurement vorgenommen. Der Entity-Typ Sample enthält die `Sample Number` als Attribut und Schlüssel, der Entity-Typ Measurement ist funktional abhängig von Sample und enthält die Attribute `Measurement Date Time` und `Record Number` als Teilschlüssel, sowie `Operator Name`.

4.1.5 Metadaten

Der Ordner Metadata enthält eine Vielzahl von Metadaten zu der Forschungsfahrt, von der die Daten des Atlas Projekts aus Abschnitt 4.1.3 stammen. Die Daten liegen in verschiedenen Dateiformaten wie .shp, .xml oder .dbf vor. Die meisten dieser Dateien können nur mit spezieller Geoinformationssystem-Software wie ArcGIS geöffnet werden und enthalten unter anderem geografische Karten, auf denen die Route der Forschungsfahrt und die Entnahmestellen der Proben aus dem Atlas Projekt eingezeichnet sind. Die Datei `PG_alle_Stationen_Sedimente.dbf` (siehe Abbildung 4.10) lässt sich jedoch auch als Tabelle in z.B. Excel oder LibreOffice öffnen und enthält weitere Informationen über die Proben der

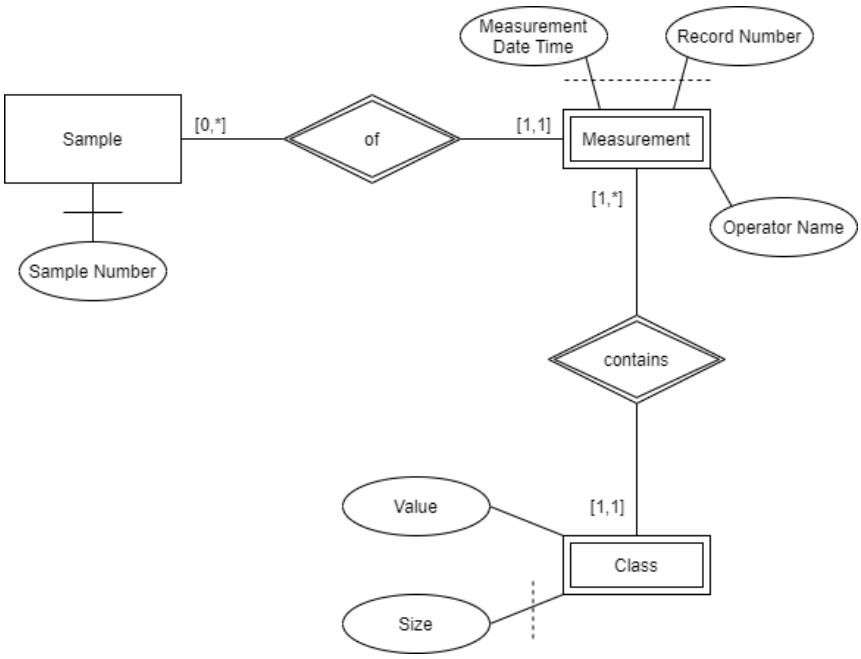

Abbildung 4.9: ER-Modell für die Mastersizer-Daten

Forschungsfahrt wie z.B. die Koordinaten, an denen die Proben entnommen wurden und eine erste grobe Einschätzung der enthaltenen Sedimentarten.

	A	B	C	D	E	F
1	ID_Name,C,254	Long,N,19,11	Lat,N,19,11	Name,	SAMPLE_TYP,C,254	TEXTURAL_G,C,254
2	ATLAS_PG01	12.75203300000	54.51242800000	PG01	Unimodal, Well Sorted	Slightly Gravelly Sand
3	ATLAS_PG02	12.79137700000	54.51400500000	PG02	Unimodal, Moderately Sorted	Sand
4	ATLAS_PG03	12.81076800000	54.51371300000	PG03	Unimodal, Moderately Sorted	Slightly Gravelly Sand
5	ATLAS_PG04	12.84145300000	54.51974300000	PG04	Unimodal, Well Sorted	Sand
6	ATLAS_PG05	12.85270300000	54.52222800000	PG05	Polymodal, Very Poorly Sorted	Gravelly Sand
7	ATLAS_PG06	12.87581700000	54.51609700000	PG06	Polymodal, Very Poorly Sorted	Sandy Gravel
8	ATLAS_PG07	12.89523500000	54.53256200000	PG07	Polymodal, Poorly Sorted	Sandy Gravel
9	ATLAS_PG08	12.89630800000	54.53559200000	PG08		No Sample
10	ATLAS_PG09	12.90012300000	54.53815500000	PG09		No Sample
11	ATLAS_PG10_aero	12.83451300000	54.53425800000	PG10	Bimodal, Poorly Sorted	Slightly Gravelly Sand
12	ATLAS_PG11	12.88417000000	54.55036700000	PG11		No Sample
13	ATLAS_PG12	12.88152800000	54.55893200000	PG12	Unimodal, Moderately Well Sorted	Slightly Gravelly Sand

Abbildung 4.10: Ausschnitt aus `PG_alle_Stationen_Sedimente.dbf`

Im Ordner AT2020 befindet sich außerdem eine Datei mit ähnlichen Metadaten zu der Forschungsfahrt und den Proben, aus denen die AT2020-Daten stammen. Die Datei `SGE_Arne2020.xlsx` ist eine tabellarische Datei mit mehreren Reitern. Im Folgenden werden nur die Reiter beschrieben, die für die Bachelorarbeit relevant sind. Der Reiter „CruiseInfo“ enthält Informationen über die Forschungsfahrt, „Sample-Description“ enthält eine erste makroskopische Beschreibung der Proben direkt nach deren Entnahme, der Reiter „Grabstation“ (zu sehen in Abbildung 4.11) enthält die Koordinaten, Wassertiefe, Zeitstempel und Wetterbedingungen für jede Probenentnahme der Fahrt.

4.2 Entwurf des ER-Modells

Nachdem im vorherigen Abschnitt die Dateistrukturen der einzelnen Ordner analysiert und durch ER-Modelle dargestellt wurden, werden in diesem Abschnitt die verschiedenen ER-Modelle zu einem Gesamtmodell zusammengeführt. Dies ermöglicht die Speicherung aller Daten in einer gemeinsamen Datenbank. Somit können Daten, die bisher in unterschiedlichen Strukturen, Dateiformaten und Speicherorten aufbewahrt wurden, mit geringem Aufwand gemeinsam abgerufen und analysiert werden.

	A	F	G	H	I	J	K	L
1	XNameGS	XPositioning	XSourceRefSystem	XXCoordinate	XYCoordinate	XSamplingDate	XTimeFormat	XDepth
2	AT2020_01	8	2	13,98375	54,66107	09.07.2020 7:03	6	23,50
3	AT2020_02	8	2	14,00982	54,63567	09.07.2020 7:34	6	23,50
4	AT2020_03	8	2	14,02550	54,64317	09.07.2020 7:53	6	22,30
5	AT2020_04	8	2	14,05483	54,66567	09.07.2020 8:18	6	23,60
6	AT2020_05	8	2	14,05450	54,70200	09.07.2020 8:44	6	24,50
7	AT2020_06	8	2	14,06267	54,71300	09.07.2020 9:00	6	24,60
8	AT2020_07	8	2	14,01167	54,69767	09.07.2020 9:25	6	23,50
9	AT2020_08	8	2	13,97683	54,68817	09.07.2020 10:02	6	25,40
10	AT2020_09	8	2	13,96683	54,70317	09.07.2020 10:21	6	27,00
11	AT2020_10	8	2	13,99350	54,71967	09.07.2020 10:42	6	26,30
12	AT2020_11	8	2	14,03067	54,73833	09.07.2020 11:06	6	26,20
13	AT2020_12	8	2	14,01383	54,76383	09.07.2020 11:45	6	35,70

Abbildung 4.11: Ausschnitt aus `SGE_Arne2020.xlsx`

Das Gesamtmodell ist in Abbildung 4.12 zu sehen. Hier werden die Daten wie in Abschnitt 4.1.4 auf drei Entity-Typen aufgeteilt: Probe, Messung und Klasse. Zusätzlich werden mittels Ist-Beziehungen spezialisierte Entity-Typen eingefügt. Diese enthalten Informationen, die nur ein Teil der Datensätze besitzt. Der Entity-Typ Altdaten ist eine Spezialisierung von Probe, die Entity-Typen Sieben und Mastersizer sind Spezialisierungen von Messung. Im Folgenden werden die einzelnen Entity-Typen genauer erklärt.

Der Entity-Typ Probe enthält alle Informationen über eine Probe, die nicht in Zusammenhang mit Messungen von dieser Probe stehen und die für die meisten Datensätze verfügbar sind. Dazu gehören die folgenden Attribute:

- `Probennummer`: Die Probennummer dient zur Identifizierung von Proben und ist in allen Datenstrukturen außer den Altdaten vorhanden. Da die Altdaten jedoch nur einen kleinen Teil aller Daten ausmachen, wird die Probennummer trotzdem als Schlüssel verwendet und für die Altdaten künstlich eine fortlaufende Nummer als Probennummer eingefügt.
- `Fahrtnummer`: Die Fahrtnummer dient zur Identifizierung der Forschungsfahrt, auf der die Probe genommen wurde. Sie ist oft Teil der Probennummern oder kann über Metadaten ermittelt werden.
- `Zeitstempel`: Das Attribut Zeitstempel enthält den Zeitpunkt, an dem die Probe genommen wurde. Diese Information kann oft über Metadaten ermittelt werden.
- `Longitude und Latitude`: Die Attribute Longitude und Latitude enthalten die Koordinaten, an denen die Probe genommen wurde. Sie können oft mithilfe von Metadaten ermittelt werden.
- `Tiefe`: Das Attribut Tiefe enthält die Wassertiefe, an der die Probe genommen wurde.
- `Bordbeschreibung und Fotodokumentation`: Diese Attribute enthalten Informationen darüber, ob eine Bordbeschreibung und Fotodokumentation der Probenentnahme vorhanden sind, und ggf. wo sich diese befinden.
- `Quelle`: Das Attribut Quelle stammt aus den Altdaten, siehe Kapitel 4.1.1. Es wurde bei der Ausgliederung der Altdaten-spezifischen Attribute in den spezialisierten Entity-Typ Altdaten entschieden, dieses Attribut in beide Entity-Typen aufzunehmen. Im Entity-Typ Altdaten enthält das Attribut die Quelle des Tupels, wie es in der Datei `all_cum.txt` enthalten ist. Im Entity-Typ Probe können dort weitere Informationen über den Ursprung des Tupels eingetragen werden.

Der Entity-Typ Altdaten ist eine Spezialisierung des Entity-Typs Probe. Er enthält alle Informationen über Proben, die nur in den Altdaten vorhanden sind. Dabei handelt es sich um die folgenden Attribute:

- `idx_map`: Das Attribut idx_map enthält vermutlich eine Art von Proben-ID, die jedoch nicht immer eindeutig vergeben wurde.

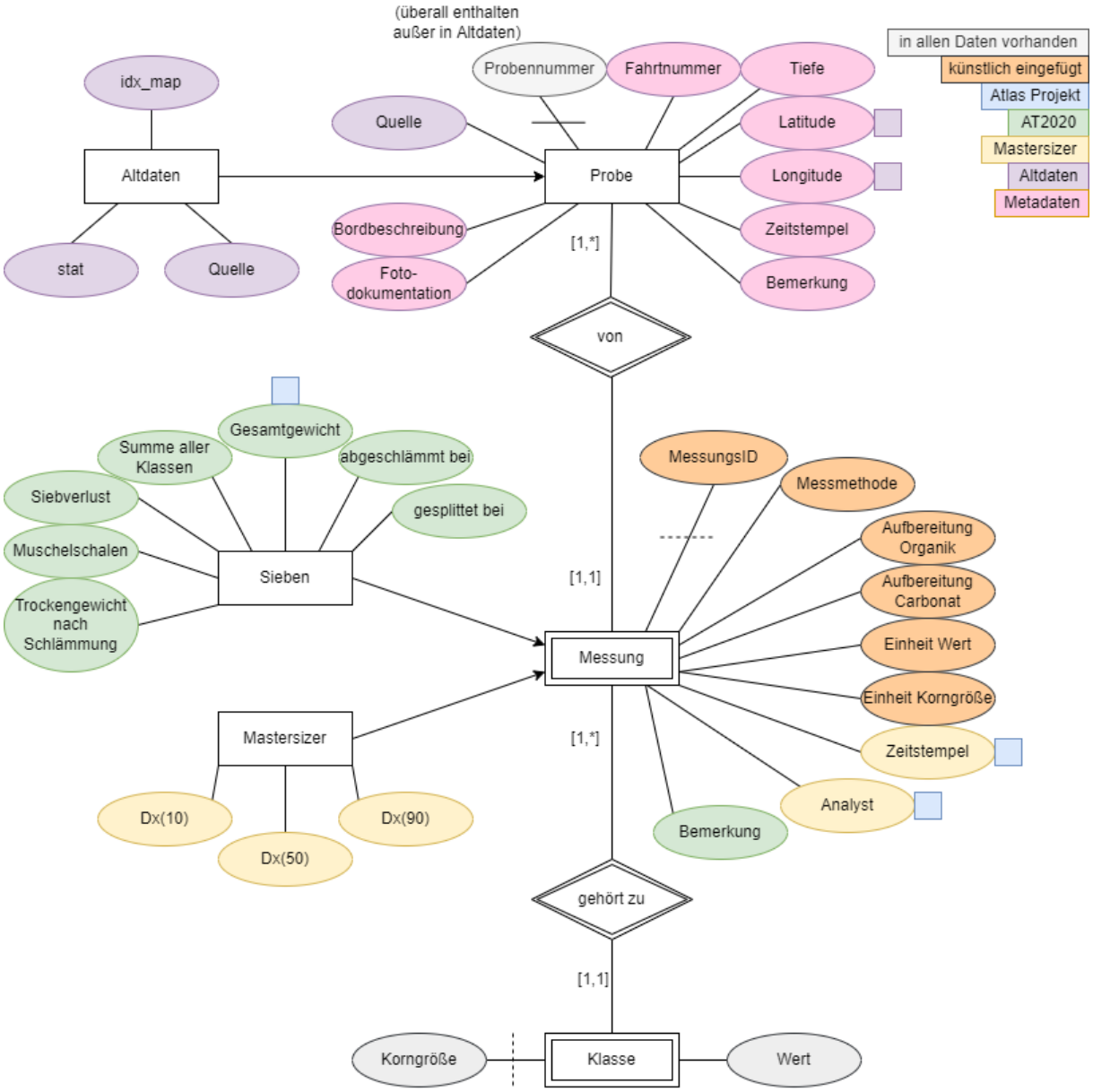

Abbildung 4.12: Gesamtmodell für alle Daten, Ursprung der Attribute farblich markiert, Attribute mit mehreren Ursprüngen sind durch Kästchen in der Farbe des anderen Ursprungs gekennzeichnet

- `stat`: Das Attribut stat enthält vermutlich eine Art von Stations-ID.
- `Quelle`: Das Attribut Quelle enthält den Ursprung des Tupels, da die Altdaten eine Sammlung von Daten verschiedener Institute sind.

Der Entity-Typ Messung enthält alle Informationen zu einer Messung einer Probe, die für die meisten Datensätze verfügbar sind, außer den Klassengrößen und deren zugehörige Messwerte. Messung ist funktional von Probe abhängig, da eine Messung nicht ohne die Probe existieren kann, an der die Messung durchgeführt wird. Der Entity-Typ enthält die folgenden Attribute:

- `Zeitstempel` und `Analyst`: Diese Attribute stammen aus den Mastersizer-Daten und dem Atlas Projekt.
- `Messmethode`, `Aufbereitung Organik` und `Aufbereitung Carbonat`: Diese Attribute sind nicht explizit in den Daten vorhanden, sondern wurden zur besseren Dateninterpretation auf Wunsch des IOW hinzugefügt.
- `Einheit Wert` und `Einheit Korngröße`: Diese Attribute wurden künstlich eingefügt, da die Messergebnisse aus den verschiedenen Strukturen in unterschiedlichen Einheiten vorliegen und aus den reinen Messwerten und Klassengrößen nicht ersichtlich ist, um welche Einheiten es sich handelt. Da sich die Einheiten innerhalb einer Messung nicht ändern, können diese als Attribute des Entity-Typs Messung gespeichert werden.
- `Bemerkung`: Dieses Attribut stammt aus den AT2020-Daten. Es wurde jedoch nicht in den Entity-Typ Sieben ausgelagert, da es für alle Messungen nützlich ist, Bemerkungen und Kommentare einfügen zu können.
- `MessungsID`: Die `MessungsID` wurde als künstlicher Schlüssel eingefügt, da es zwischen den verschiedenen Datenstrukturen keine einheitlich zu bestimmende Attributmenge gibt, die sich als Schlüssel eignet. Da Messung von Probe abhängig ist und es sich somit bei `MessungsID` um einen Teilschlüssel in Kombination mit `Probennummer` handelt, kann je Instanz von Probe die `MessungsID` als eine wieder von Beginn zählende fortlaufende Nummer implementiert werden.

Der Entity-Typ Sieben ist eine Spezialisierung von Messung. Er enthält die Informationen, die nur für Messungen durch Sieben vorhanden sind. Dazu zählen die folgenden Attribute:

- `gesplittet bei` und `abgeschlämmt bei`: Diese Attribute stammen aus den AT2020-Daten, wurden jedoch so angepasst, dass die Grenze flexibel angegeben werden kann. In den Originaldaten ist die Grenze festgelegt und es wird nur ja bzw. nein eingetragen.
- `Gesamtgewicht`: Das Attribut `Gesamtgewicht` fasst das Gesamtgewicht aus AT2020 und Sample Weight aus dem Atlas Projekt zusammen.
- `Summe aller Klassen`, `Siebverlust`, `Muschelschalen`, `Trockengewicht nach Schlämmung`: Diese Attribute wurden aus den AT2020-Daten übernommen.

Der Entity-Typ Mastersizer ist eine Spezialisierung von Messung. Er enthält die Informationen, die nur für Messungen vorhanden sind, die mit dem Mastersizer durchgeführt wurden. Dabei handelt es sich um die statistischen Auswertungsergebnisse `Dx(10)`, `Dx(50)` und `Dx(90)`.

Der Entity-Typ Klasse enthält die Messwerte der Korngrößenklassen. Klasse ist funktional abhängig von Messung, da die Messwerte in Klasse nur im Rahmen einer Messung entstehen können. Der Entity-Typ enthält zwei Attribute:

- `Korngröße`: Das Attribut `Korngröße` enthält die Untergrenze der Korngrößenklasse. `Korngröße` ist auch Teilschlüssel von Klasse, da pro Messung jede Korngrößenklasse nur einmal gemessen wird.

- `Wert`: Das Attribut `Wert` enthält den Messwert, der zur zugehörigen Korngröße ermittelt wurde.

4.3 Erweiterung des Entwurfs um Provenance

Das in Abschnitt 4.2 erstellte Gesamtmodell soll nun so erweitert werden, dass zusätzliche Informationen gespeichert werden können, die die Umsetzung von Provenancetechniken ermöglichen. Das Hauptziel liegt dabei in der Umsetzung von Data Provenance. Im Folgenden werden die Elemente erläutert, die dem Modell hinzugefügt wurden. Das um Provenance erweiterte ER-Modell ist in Abbildung 4.13 zu sehen.

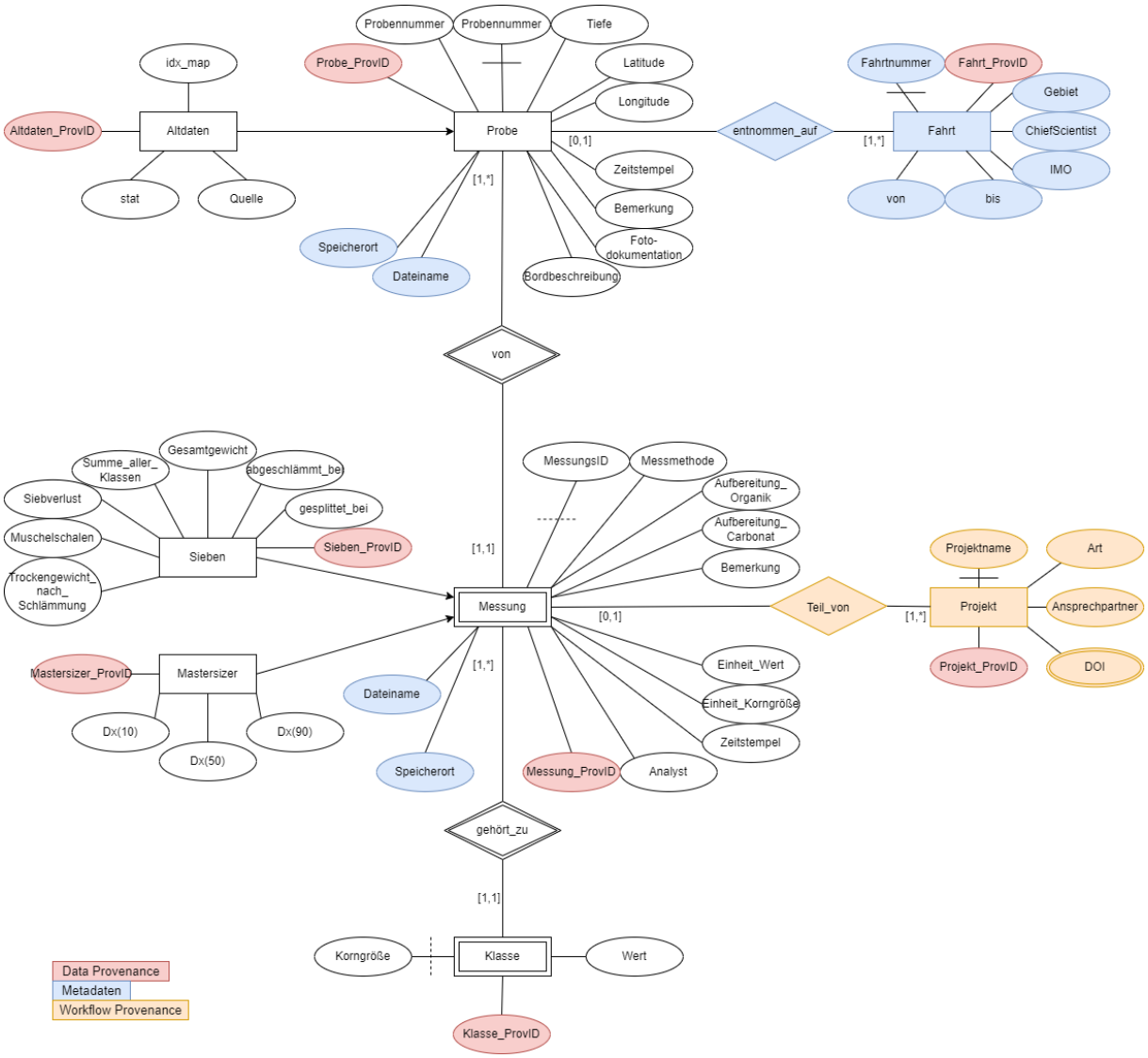

Abbildung 4.13: Um Provenance erweitertes Gesamtmodell für alle Daten

Zunächst werden weitere Metadaten über die Forschungsfahrt, von der eine Probe stammt, ins Model aufgenommen. Hierzu wird der Entity-Typ `Fahrt` erstellt und über den Beziehungstyp `entnommen_auf` mit `Probe` verbunden. Auf einer Forschungsfahrt können viele Proben entnommen werden und Fahrten, auf denen keine Sedimentproben entnommen wurden, sind für die Datenbank nicht relevant, also erhält `Fahrt` das Kardinalitätsintervall [1,*]. Es ist nicht zwingend für jede Probe bekannt, von welcher Fahrt sie stammt, und eine Probe kann nur von maximal einer Fahrt stammen. Also erhält `Probe` das Intervall [0,1]. Zudem erhält `Fahrt` die folgenden Attribute:

- `Fahrtnummer`: Jede Forschungsfahrt am IOW hat eine zugeordnete Fahrtnummer, die die Fahrt eindeutig identifiziert. Diese wird somit zum Schlüssel. Vor der Erweiterung des Modells war die Fahrtnummer als Attribut von `Probe` enthalten, dort kann das Attribut nun also entfernt werden.
- `Gebiet`: Dieses Attribut enthält das Meeresgebiet, in dem sich die Forschungsfahrt bewegt hat.
- `ChiefScientist`: Dieses Attribut enthält den Wissenschaftler, der für die Sedimentproben auf der Fahrt verantwortlich war.
- `IMO`: Dieses Attribut enthält die IMO-Nummer des Forschungsschiffs. Es handelt sich dabei um eine international eindeutige Identifikationsnummer für Schiffe der Internationalen Seeschifffahrts-Organisation IMO.
- `von` und `bis`: Diese Attribute enthalten das Start- und Enddatum der Fahrt.

Im Sinne der Provenance ist es für die Transparenz und Nachvollziehbarkeit von Forschungsergebnissen wichtig, dass dokumentiert wird, welche Daten in einem Forschungsprojekt verwendet wurden. Hierfür wird ein neuer Entity-Typ `Projekt` erstellt. Er ist über den Beziehungstyp `Teil_von` mit `Messung` verbunden. Projekte, an denen keine Sedimentdaten beteiligt waren, sind für die Datenbank nicht relevant und im Rahmen eines Projekts werden in der Regel viele Messungen vorgenommen, also erhält `Projekt` das Kardinalitätsintervall [1,*]. Eine Messung wird im IOW nicht für mehrere Projekte erhoben, und insbesondere bei den Altdaten ist nicht zwingend bekannt, in welchem Projekt die Daten verwendet wurden. Daher erhält `Messung` das Intervall [0,1]. `Projekt` enthält die folgenden Attribute:

- `Projektname`: Dieses Attribut enthält einen eindeutigen Projektnamen und dient als Schlüssel.
- `Art`: In diesem Attribut kann beschrieben werden, um was für ein Projekt es sich handelt.
- `Ansprechpartner`: Dieses Attribut enthält einen Verweis auf die Person, die für das Projekt verantwortlich ist und ggf. Fragen beantworten könnte. Hier könnte entweder der Name der Person gespeichert werden, oder die WissenschaftlerID aus der IOWDB[2], der zentralen Forschungsdatenbank des IOW.
- `DOI`: Bei dem Attribut `DOI` handelt es sich um ein mengenwertiges Attribut, in dem die Digital Object Identifiers von allen Veröffentlichungen gespeichert werden können, die ggf. im Rahmen eines Projekts entstanden sind. Da alle anderen Informationen über eine Veröffentlichung wie Autoren, Veröffentlichungsdatum u.ä. mithilfe der DOI mit geringem Aufwand ermittelt werden können, werden in diesem Modell nur die DOIs gespeichert.

Um Data Provenance zu ermöglichen, muss jedes Tupel jeder Relation der Datenbank global eindeutig identifizierbar sein. Schlüssel müssen jedoch nur die Tupel innerhalb einer Relation eindeutig identifizieren. Daher reichen die bisher ausgewählten Schlüssel nicht für Data Provenance aus. Stattdessen werden zusätzlich Provenance-IDs vergeben, die im gesamten Modell global einzigartig sind. Hierfür erhält jeder Entity-Typ des Modells ein weiteres Attribut namens `[Entityname]_ProvID`. Mithilfe dieser IDs kann jedes Tupel eindeutig identifiziert werden und somit kann Data Provenance auf Datenbankanfragen angewendet werden, wie im Abschnitt 2.5 beschrieben.

Des Weiteren sollen zusätzliche Informationen über die Herkunft des Datensatzes gespeichert werden können. Da die Daten bisher dezentral in verschiedenen Dateien an verschiedenen Orten gespeichert sind, sollten diese Informationen im Modell mit aufgenommen werden. Dies ermöglicht das Zurückverfolgen eines Tupels zu der Datei, aus der der originale Datensatz stammt. Hierzu erhalten die Entity-Typen `Probe` und `Messung` je zwei weitere Attribute: `Dateiname` und `Speicherort`. So kann für jedes Tupel

[2] `https://odin2.io-warnemuende.de/`

dokumentiert werden, aus welcher Datei die Daten stammen und auf welchem Rechner bzw. Server und unter welchem Dateipfad diese Datei gespeichert ist. Mithilfe dieser Informationen können beispielsweise Fehler gefunden und korrigiert werden, die beim Einpflegen der Daten in die Datenbank entstanden. Der Speicherort ist zwar langfristig nur korrekt, wenn die Ursprungsdateien nicht verschoben werden oder beim Verschieben von Dateien auch deren Speicherort in der Datenbank geändert wird. Dennoch erhöht das Speichern dieser Informationen die Rückverfolgbarkeit der Datensätze.

4.4 Entwurf des relationalen Datenbankmodells

Um das ER-Modell in ein relationales Datenbankmodell zu überführen, wird zunächst für jede Entity und Beziehung eine eigene Relation mit den zugehörigen Attributen und Schlüsseln erstellt. Die Relationen der abhängigen Entitys erhalten zusätzlich als Attribut und partiellen Fremdschlüssel den Schlüssel der Entity, von der sie abhängig sind. Die Relationen der Beziehungen erhalten zusätzlich als Attribute die Schlüssel der an der Beziehung beteiligten Entitys. Die Primärschlüssel der Beziehungen müssen so gewählt werden, dass die Abbildung kapazitätserhaltend ist. Ist-Beziehungen werden nicht auf eine eigene Relation abgebildet, stattdessen erhält der speziellere Entity-Typ den Primärschlüssel des generellen Entity-Typs als Fremd- und Primärschlüssel. So entstehen folgende Relationen:

1. PROBE = {Probennummer, Zeitstempel, Tiefe, Latitude, Longitude, Quelle, Bordbeschreibung, Fotodokumentation, Bemerkung, Dateiname, Speicherort, Probe_ProvID}
   mit $\mathcal{K}_{\text{PROBE}}$ = {{Probennummer}}

2. ALTDATEN = {Probennummer, idx_map, stat, Quelle, Altdaten_ProvID}
   mit $\mathcal{K}_{\text{ALTDATEN}}$ = {{Probennummer}}

3. MESSUNG = {Probennummer, MessungsID, Messmethode, Aufbereitung_Organik, Aufbereitung_Carbonat, Einheit_Wert, Einheit_Korngröße, Zeitstempel, Analyst, Bemerkung, Dateiname, Speicherort, Messung_ProvID}
   mit $\mathcal{K}_{\text{MESSUNG}}$ = {{Probennummer, MessungsID}}

4. SIEBEN = {Probennummer, MessungsID, gesplittet_bei, abgeschlämmt_bei, Gesamtgewicht, Summe_aller_Klassen, Siebverlust, Muschelschalen, Trockengewicht_nach_Schlämmung, Sieben_ProvID}
   mit $\mathcal{K}_{\text{SIEBEN}}$ = {{Probennummer, MessungsID}}

5. MASTERSIZER = {Probennummer, MessungsID, Dx(10), Dx(50), Dx(90), Mastersizer_ProvID}
   mit $\mathcal{K}_{\text{MASTERSIZER}}$ = {{Probennummer, MessungsID}}

6. KLASSE = {Probennummer, MessungsID, Korngröße, Wert, Klasse_ProvID}
   mit $\mathcal{K}_{\text{Klasse}}$ = {{Probennummer, MessungsID, Korngröße}}

7. FAHRT = {Fahrtnummer, Gebiet, ChiefScientist, IMO, von, bis, Fahrt_ProvID}
   mit $\mathcal{K}_{\text{FAHRT}}$ = {{Fahrtnummer}}

8. PROJEKT = {Projektname, Art, Ansprechpartner, DOI, Projekt_ProvID}
   mit $\mathcal{K}_{\text{PROJEKT}}$ = {{Name}}

9. TEIL_VON = {Probennummer, MessungsID, Projektname}
   mit $\mathcal{K}_{\text{TEIL\_VON}}$ = {{Probennummer, MessungsID}}

10. ENTNOMMEN_AUF = {Probennummer, Fahrtnummer}
    mit $\mathcal{K}_{\text{ENTNOMMEN\_AUF}}$ = {{Probennummer}}

11. VON = {Probennummer, MessungsID}
    mit $\mathcal{K}_{\text{VON}}$ = {{Probennummer, MessungsID}}

12. GEHÖRT_ZU = {Probennummer, MessungsID, Korngröße}
    mit $\mathcal{K}_{\text{GEHÖRTZU}}$ = {{Probennummer, MessungsID, Korngröße}}

Im nächsten Schritt werden wenn möglich die Relationen von Beziehungstypen und Entity-Typen verschmolzen. Dies ist der Fall, wenn der Beziehungstyp durch ein [1,1]-Intervall mit dem Entity-Typen verbunden ist. Es sind in diesem Modell also folgende Verschmelzungen möglich:

- VON kann mit MESSUNG verschmolzen werden.
- GEHÖRT_ZU kann mit KLASSE verschmolzen werden
- ENTNOMMEN_AUF kann mit PROBE verschmolzen werden.

Da es sich bei MESSUNG und KLASSE um abhängige Entity-Typen handelt, enthalten sie bereits die Fremdschlüssel der zu verschmelzenden Beziehungen. Außerdem enthalten alle Beziehungen im Modell keine weiteren Attribute als die Schlüssel. Daher müssen MESSUNG und KLASSE nicht verändert werden, VON und GEHÖRT_ZU können einfach entfernt werden. Um PROBE und ENTNOMMEN_AUF zu verschmelzen, wird in PROBE der Primärschlüssel von FAHRT als Fremdschlüssel eingefügt und ENTNOMMEN_AUF wird entfernt. Nach dem Verschmelzen verbleiben die folgenden Relationen:

1. PROBE = {Probennummer, Zeitstempel, Tiefe, Latitude, Longitude, Quelle, Bordbeschreibung, Fotodokumentation, Bemerkung, Dateiname, Speicherort, Fahrtnummer, Probe_ProvID}
   mit $\mathcal{K}_{\text{PROBE}}$ = {{Probennummer}}

2. ALTDATEN = {Probennummer, idx_map, stat, Quelle, Altdaten_ProvID}
   mit $\mathcal{K}_{\text{ALTDATEN}}$ = {{Probennummer}}

3. MESSUNG = {Probennummer, MessungsID, Messmethode, Aufbereitung_Organik, Aufbereitung_Carbonat, Einheit_Wert, Einheit_Korngröße, Zeitstempel, Analyst, Bemerkung, Dateiname, Speicherort, Messung_ProvID}
   mit $\mathcal{K}_{\text{MESSUNG}}$ = {{Probennummer, MessungsID}}

4. SIEBEN = {Probennummer, MessungsID, gesplittet_bei, abgeschlämmt_bei, Gesamtgewicht, Summe_aller_Klassen, Siebverlust, Muschelschalen, Trockengewicht_nach_Schlämmung, Sieben_ProvID}
   mit $\mathcal{K}_{\text{SIEBEN}}$ = {{Probennummer, MessungsID}}

5. MASTERSIZER = {Probennummer, MessungsID, Dx(10), Dx(50), Dx(90), Mastersizer_ProvID}
   mit $\mathcal{K}_{\text{MASTERSIZER}}$ = {{Probennummer, MessungsID}}

6. KLASSE = {Probennummer, MessungsID, Korngröße, Wert, Klasse_ProvID}
   mit $\mathcal{K}_{\text{Klasse}}$ = {{Probennummer, MessungsID, Korngröße}}

7. FAHRT = {Fahrtnummer, Gebiet, ChiefScientist, IMO, von, bis, Fahrt_ProvID}
   mit $\mathcal{K}_{\text{FAHRT}}$ = {{Fahrtnummer}}

8. PROJEKT = {Projektname, Art, Ansprechpartner, DOI, Projekt_ProvID}
   mit $\mathcal{K}_{\text{PROJEKT}}$ = {{Projektname}}

9. TEIL_VON = {Probennummer, MessungsID, Projektname}
   mit $\mathcal{K}_{\text{TEIL\_VON}}$ = {{Probennummer, MessungsID}}

Nun werden die Relationenschemata auf die Erfüllung der Normalformen überprüft und wenn nötig geändert. Das Attribut `DOI` aus PROJEKT wurde im ER-Modell als mengenwertiges Attribut modelliert, was gegen die erste Normalform verstößt. Somit wird `DOI` in diesem Schritt zu einem atomaren Attribut umgeformt, dh. alle Tupel mit mehreren Elementen im Attributwert von `DOI` werden so dupliziert, dass für jedes Element des Attributs ein eigenes Tupel existiert. Dadurch ist `DOI` jedoch nicht mehr funktional abhängig vom Schlüssel `Projektname` der Relation und muss zum Schlüssel hinzugefügt werden. Durch die Erweiterung des Schlüssels von PROJEKT entstehen jedoch partielle Abhängigkeiten, die gegen die zweite Normalform verstoßen. Um die zweite Normalform zu erfüllen, wird PROJEKT auf zwei Relationen aufgeteilt, indem eine neue Relation PROJEKT_DOI erstellt wird. Die neue Relation erhält zusätzlich ein Attribut `Projekt_DOI_ProvID` für die Umsetzung von Data Provenance. Die dritte Normalform wird theoretisch in der Relation SIEBEN verletzt, da Siebverlust aus dem Gesamtgewicht und der Summer aller Klassen berechnet werden könnte, also eine transitive Abhängigkeit $\{\texttt{Probennummer}, \texttt{MessungsID}\} \rightarrow \{\texttt{Gesamtgewicht}, \texttt{Summe\_aller\_Klassen}\} \rightarrow \texttt{Siebverlust}$ existiert. In der Praxis ist diese funktionale Abhängigkeit jedoch nicht zwingend gegeben, da in den Ursprungsdateien Rechenfehler vorhanden sein können. Im Sinne der Provenance sollten diese dokumentiert werden. Daher wird diese potentielle transitive Abhängigkeit nicht behoben, sondern verbleibt als Kontrollmechanismus im Relationenmodell. Zuletzt werden Attributnamen, die in mehreren Relationen vorkommen und keine Fremdschlüssel sind, umbenannt, indem der Relationenname vor den Attributnamen gesetzt wird. Außerdem werden Umlaute entfernt. Somit besteht das Datenbankschema aus folgenden Relationenschemata:

1. PROBE = {Probennummer, Probe_Zeitstempel, Tiefe, Latitude, Longitude, Probe_Quelle, Bordbeschreibung, Fotodokumentation, Probe_Bemerkung, Probe_Dateiname, Probe_Speicherort, Fahrtnummer, Probe_ProvID}
   mit $\mathcal{K}_{\text{PROBE}}$ = {{Probennummer}}

2. ALTDATEN = {Probennummer, idx_map, stat, Altdaten_Quelle, Altdaten_ProvID}
   mit $\mathcal{K}_{\text{ALTDATEN}}$ = {{Probennummer}}

3. MESSUNG = {Probennummer, MessungsID, Messmethode, Aufbereitung_Organik, Aufbereitung_Carbonat, Einheit_Wert, Einheit_Korngroesse, Messung_Zeitstempel, Analyst, Messung_Bemerkung, Messung_Dateiname, Messung_Speicherort, Messung_ProvID}
   mit $\mathcal{K}_{\text{MESSUNG}}$ = {{Probennummer, MessungsID}}

4. SIEBEN = {Probennummer, MessungsID, gesplittet_bei, abgeschlaemmt_bei, Gesamtgewicht, Summe_aller_Klassen, Siebverlust, Muschelschalen, Trockengewicht_nach_Schlaemmung, Sieben_ProvID}
   mit $\mathcal{K}_{\text{SIEBEN}}$ = {{Probennummer, MessungsID}}

5. MASTERSIZER = {Probennummer, MessungsID, Dx(10), Dx(50), Dx(90), Mastersizer_ProvID}
   mit $\mathcal{K}_{\text{MASTERSIZER}}$ = {{Probennummer, MessungsID}}

6. KLASSE = {Probennummer, MessungsID, Korngroesse, Wert, Klasse_ProvID}
   mit $\mathcal{K}_{\text{Klasse}}$ = {{Probennummer, MessungsID, Korngroesse}}

7. FAHRT = {Fahrtnummer, Gebiet, ChiefScientist, IMO, von, bis, Fahrt_ProvID}
   mit $\mathcal{K}_{\text{FAHRT}}$ = {{Fahrtnummer}}

8. PROJEKT = {Projektname, Art, Ansprechpartner, Projekt_ProvID}
   mit $\mathcal{K}_{\text{PROJEKT}}$ = {{Projektname}}

9. TEIL_VON = {Probennummer, MessungsID, Projektname}
   mit $\mathcal{K}_{\text{TEIL\_VON}}$ = {{Probennummer, MessungsID}}

10. PROJEKT_DOI = {Projektname, DOI, Projekt_DOI_ProvID}
    mit $\mathcal{K}_{\text{PROJEKT\_DOI}}$ = {{Projektname, DOI}}

4.5 Einpflegen der Daten

Wenn die entworfene Datenbank erstellt wurde, müssen als Nächstes die Daten importiert werden. Die Datenbank wurde so entworfen, dass sie umfangreich und universell genug ist, um für alle vorhandenen Datenstrukturen nutzbar zu sein und zusätzliche Metadaten aufzunehmen. Dies führt jedoch zu dem Nachteil, dass die bestehenden Daten nicht einfach in ihrem ursprünglichen Format eingelesen und in die Datenbank eingepflegt werden können. Stattdessen müssen die Originaldateien angepasst werden, um sie auf den Import in die Datenbank vorzubereiten. Um die Originaldateien in ihrem ursprünglichen Zustand zu erhalten, sollten alle Anpassungen grundsätzlich nur auf Kopien der Dateien vorgenommen werden. Die Änderungen, die in dieser Arbeit beschrieben werden, sind auf den Datenimport mithilfe des Table Data Import Wizards von MySQL ausgelegt. Wenn die Daten auf eine andere Art in die Datenbank importiert werden sollen, sind ggf. andere Änderungen nötig.

Da die Strukturen der Originaldaten so divers sind, kann keine allgemeingültige Aussage über alle nötigen Änderungen für jede am IOW vorhandene Datei mit Sedimentdaten gemacht werden. In Abschnitt 5.2 werden jedoch die nötigen Änderungen für alle Beispieldatensätze detailliert erläutert, die für diese Arbeit vom IOW zur Verfügung gestellt wurden. An dieser Stelle werden nur die generellen Voraussetzungen beschrieben, die erfüllt sein müssen, um Daten mit dem Table Data Import Wizard in die Datenbank einzupflegen:

- Teilweise sind zusammengehörige Informationen auf mehrere Dateien verteilt. Beispielsweise können Informationen über eine Forschungsfahrt und die dort entnommenen Proben in einer Datei gespeichert sein, und die Informationen über die Messungen dieser Proben in einer oder mehreren anderen Dateien. Es müssen dann alle Dateien mit relevanten Informationen separat angepasst und importiert werden.

- Zu importierende Dateien müssen so aufgebaut sein, dass eine Spalte einem Attribut entspricht und eine Zeile einem Datensatz, wobei die erste Zeile die Spaltenüberschriften enthält. Spaltenüberschriften sollten in die entsprechenden Attributnamen der Datenbank umbenannt werden. Um eine Datei mit vertauschten Achsen zu importieren, muss diese zunächst in die richtige Form umformatiert werden.

- Leerzeilen zwischen Datensätzen sollten entfernt werden.

- MessungsIDs müssen manuell vergeben und in die Datei eingefügt werden, wobei es sich um eine fortlaufende Nummer in Abhängigkeit von der Probennummer handeln muss. D.h. die erste Messung einer Probe erhält die MessungsID 1, die zweite Messung derselben Probe die MessungsID 2, die erste Messung einer anderen Probe die MessungsID 1 usw.

- Für jede Relation, in die Daten aus der Datei importiert werden sollen, sollte eine eigene Datei erstellt werden, in die nur die für diese Relation relevanten Spalten übernommen werden. Relationenübergreifende Spalten wie Fremdschlüssel müssen in jeder relevanten Datei vorkommen, d.h. die MessungsIDs sollten vor dem Aufteilen in separate Dateien vergeben werden.

- Datums- und Zeitangaben müssen dem Format YYYY-MM-DD hh:mm:ss entsprechen und ggf. umformatiert werden, wenn sie nicht bereits in diesem Format vorliegen.

- Die Attribute `abgeschlämmt_bei` und `gesplittet_bei` der Relation SIEBEN wurden bewusst so gewählt, dass der genaue Wert gespeichert wird, bei dem abgeschlämmt bzw. gesplittet wurde. In den Dateien wird stattdessen meist der Wert in der Spaltenüberschrift angegeben und in den Datensätzen nur „ja“ oder „nein“ eingetragen. Daher müssen in diesen Spalten die Zellen, in denen „ja“ steht, durch den Wert aus der Spaltenüberschrift ersetzt werden.

- Bei einigen Attributen der Datenbank handelt es sich um Metadaten, die meist nicht explizit in den Daten vorhanden sind und manuell eingefügt werden müssen. Daher sollte vor dem Import einer Datei in eine Relation noch einmal überprüft werden, welche Attribute der Relation nicht in der Datei vorkommen und ob diese vor dem Import noch in die Datei eingefügt werden können. Hier sind insbesondere die Attribute `Messmethode, Aufbereitung_Organik, Aufbereitung_Carbonat, Einheit_Wert` und `Einheit_Korngröße` der Relation MESSUNG zu beachten. Wenn solche Informationen bekannt oder auffindbar sind, sollten diese noch als zusätzliche Spalten eingefügt werden.

- Da die Menge der untersuchten Korngrößenklassen und die genauen Klassengrenzen sehr unterschiedlich sind, wird jedes Wertepaar aus Korngröße und Messwert als ein eigenes Tupel in der Datenbank gespeichert. In den Ursprungsdateien sind jedoch meist alle Messwerte einer Messung in einer Zeile gespeichert, mit den Korngrößen als Spaltenüberschriften oder auch in der Zeile. Um die Wertepaare in die Datenbank zu importieren, muss die Datei so umformatiert werden, dass jedes Wertepaar in einer eigenen Zeile steht, mit einer Spalte für alle Korngrößen und einer Spalte für alle Messwerte.

- Für die Relationen Probe und Messung sollten zusätzliche Spalten eingefügt werden, in denen für jede Zeile der Dateiname und der Speicherort der Originaldatei eingetragen werden, aus der der Datensatz stammt.

- Als letzter Schritt wird in jede Datei eine Spalte `Relationenname_ProvID` eingefügt und jeder Zeile eine Provenance-ID in der Form „RelationennameX“ zugewiesen.

- Alle Dateien, die in die Datenbank importiert werden sollen, müssen im .csv-Format gespeichert werden.

- Aufgrund der Fremdschlüsselbedingungen zwischen den Relationen ist dringend auf die richtige Reihenfolge beim Import zu achten. In eine Relation mit Fremdschlüsseln aus einer anderen Relation können nur Tupel importiert werden, wenn in der anderen Relation bereits die Tupel mit den Werten vorhanden sind, auf die der Fremdschlüssel verweist. So können beispielsweise keine Daten in MESSUNG importiert werden, solange die Proben, von denen die Messungen stammen, noch nicht in der Relation PROBE existieren.

In diesem Kapitel wurde ein Datenbankentwurf für die Sedimentdaten erstellt und es wurde erörtert, welche Form die Daten annehmen müssen, um in die Datenbank eingepflegt werden zu können. Im folgenden Kapitel wird der Datenbankentwurf mit MySQL implementiert, die vorhandenen Datensätze angepasst und in die erstellte Datenbank importiert.

5 Implementierung

Dieses Kapitel beschreibt die Implementierung der Datenbank, die im letzten Kapitel entworfen wurde. Hierzu wird erst die Datendefinitionssprache von SQL genutzt, um die Datenbank zu erstellen. Danach werden die Beispieldaten auf den Import vorbereitet und in die erstellte Datenbank eingefügt.

5.1 Erstellen der Datenbank

Zum Erstellen der Datenbank werden die DDL-Statements von SQL genutzt, die in Abschnitt 2.4 beschrieben wurden. Als Grundlage dienen die in Abschnitt 4.4 definierten Relationenschemata. An dieser Stelle werden nur beispielhaft einige DDLs aufgeführt, die gesamte Liste aller DDLs zur Erstellung der Datenbank ist im Anhang der Arbeit zu finden.

Bei der Erstellung der Relationen muss auf die richtige Reihenfolge geachtet werden, damit Fremdschlüsselbedingungen definiert werden können. Wenn eine Relation erstellt werden soll, die einen Fremdschlüssel enthält, muss die Relation aus der der Fremdschlüssel stammt bereits existieren. Alternativ können die Relationen auch ohne Fremdschlüsselbedingungen in beliebiger Reihenfolge erstellt werden und die Fremdschlüsselbedingungen nach Erstellung aller Relationen nachträglich eingefügt werden. Hier wurde sich dafür entschieden, die Fremdschlüssel während des Erstellens der Relationen zu definieren. Dazu wurden die Relationen in der Reihenfolge FAHRT, PROBE, ALTDATEN, MESSUNG, SIEBEN, MASTERSIZER, KLASSE, PROJEKT, TEIL_VON, PROJEKT_DOI erstellt.

Dabei muss für jedes Attribut ein passender Datentyp definiert werden. Für IDs ist beispielsweise der Datentyp INT UNSIGNED geeignet, in dem ganze, nicht negative Zahlen gespeichert werden. Für Dezimalzahlen kann FLOAT verwendet werden, Zeichenketten können speicherplatzsparend als VARCHAR gespeichert werden. Für Datums- und/oder Zeitangaben eignen sich die Datentypen DATE, TIME oder DATETIME. Manche Attribute, die einen begrenzten Wertebereich haben, können als ENUM gespeichert werden, in dem alle möglichen Attributwerte definiert sind. Die Verwendung der Datentypen ENUM und DATE/TIME/DATETIME ist, wenn möglich, besonders empfehlenswert, da sie Fehler bei der Datenintegration vorbeugen. So können keine Werte in die Datenbank eingefügt werden, die nicht dem erwarteten Format bzw. Wertebereich entsprechen. Stattdessen wird von MySQL eine Fehlermeldung ausgegeben, woraufhin der fehlerhafte Wert überprüft und korrigiert werden muss, bevor das Tupel in die Datenbank eingefügt werden kann. Dies hat unter anderem den Vorteil, dass bei Datenbankanfragen auf einen bestimmten Attributwert mit solchen Datentypen sicher alle gewünschten Tupel im Ergebnis enthalten sind. Wenn stattdessen beispielsweise VARCHAR als Datentyp gewählt wurde, wären eigentlich gewünschte Tupel mit Tippfehlern nicht im Anfrageergebnis enthalten.

In Anweisung 5.1 sind die DDLs für die Relationen PROBE und MESSUNG dargestellt, wobei MESSUNG mehrere Attribute vom Datentyp ENUM besitzt. So stehen beispielsweise für das Attribut `Messmethode` die Zeichenketten „sieben“, „mastersizer_wet“ und „mastersizer_aero“ als Wertebereich zur Verfügung. Wenn ein Datenbanknutzer nun versucht, ein Tupel mit einer anderen Messmethode oder einem Schreibfehler in der Messmethode in die Datenbank einzufügen, fügt SQL das Tupel nicht ein und gibt stattdessen

```
CREATE TABLE probe (
Probennummer VARCHAR(30),
Probe_Zeitstempel DATETIME,
Tiefe FLOAT,
Latitude FLOAT,
Longitude FLOAT,
Probe_Quelle VARCHAR(255),
Bordbeschreibung VARCHAR(150),
Fotodokumentation VARCHAR(150),
Probe_Bemerkung VARCHAR(500),
Fahrtnummer VARCHAR(20),
PRIMARY KEY(Probennummer),
FOREIGN KEY(Fahrtnummer) REFERENCES fahrt(Fahrtnummer)
);

CREATE TABLE messung(
Probennummer VARCHAR(30),
MessungsID INT UNSIGNED NOT NULL,
Messmethode ENUM('sieben','mastersizer_wet','mastersizer_aero'),
Aufbereitung_Organik VARCHAR(20),
Aufbereitung_Carbonat VARCHAR(20),
Einheit_Wert ENUM('kumulativ','prozent','gramm','milligramm'),
Einheit_Korngroesse ENUM('millimeter','mikrometer'),
Messung_Zeitstempel DATETIME,
Analyst VARCHAR(50),
Messung_Bemerkung VARCHAR(255),
PRIMARY KEY(Probennummer, MessungsID),
FOREIGN KEY(Probennummer) REFERENCES probe(Probennummer)
);
```

Anweisung 5.1: DDLs für PROBE und MESSUNG

eine Fehlermeldung aus. Falls in Zukunft eine weitere Messmethode am IOW benutzt wird, kann diese nachträglich zum Wertebereich hinzugefügt werden.

5.2 Einpflegen der Daten

Nachdem alle benötigten Relationen implementiert wurden, können die Sedimentdaten in die Datenbank eingefügt werden. Die INSERT INTO Anweisung von SQL eignet sich dazu, einzelne Tupel wie in Anweisung 5.2 oder eine kleine Anzahl von Tupeln manuell in die Datenbank einzufügen. Das manuelle Einfügen von großen Mengen an Tupeln ist jedoch sehr zeitaufwändig und fehleranfällig und sollte soweit wie möglich automatisiert werden. Die MySQL Workbench bietet hierzu den Table Data Import Wizard an, mit dem Datensätze aus .csv-Dateien in eine Relation importiert werden können.

```
INSERT INTO fahrt (Fahrtnummer, Gebiet, ChiefScientist, IMO, von, bis) VALUES ('AT2020','
    Baltic␣Sea,␣Mecklenburg␣Bight,␣Kadetrinne,␣western␣Roennebank','Dr.␣Svenja␣Papenmeier
    ','7517624', '2020-6-29', '2020-7-10');
```

Anweisung 5.2: INSERT INTO Anweisung um ein Tupel in die Relation FAHRT einzufügen

Um den Table Data Wizard zu nutzen, müssen die Originaldateien jedoch oft noch modifiziert werden. Hier wurde der Ansatz gewählt, für jede Relation die relevanten Spalten bzw. Zeilen aus der Originaldatei in eine eigene Datei zu kopieren und diese entsprechend zu bearbeiten. Je nach Struktur der Originaldatei und Zielrelation sind andere Modifikationen nötig, die in den folgenden Abschnitten für die bisher behandelten Beispieldaten beschrieben werden. Zum Modifizieren der Dateien wird in dieser Arbeit das

Programm LibreOffice und dessen Bearbeitungsfunktionen genutzt. Ähnliche Funktionen sind in der Regel auch in vergleichbaren Programmen wie MS Excel oder OpenOffice verfügbar. Hier wird jedoch nur die Vorgehensweise für die Arbeit mit LibreOffice beschrieben.

Sortieren von Tabellen Um eine Tabelle zu sortieren, wird zunächst der zu sortierende Bereich ausgewählt. Dabei sollte die erste Zeile der Tabelle ausgelassen werden, da sie Spaltenüberschriften enthält. Dann wird im Menü „Daten → Sortieren“ ausgewählt. Es öffnet sich ein Popup-Menü, in dem die Sortierkriterien festgelegt werden müssen. Es können bis zu 3 Spalten ausgewählt werden, nach denen sortiert wird, und ob die Sortierung für die jeweilige Spalte aufsteigend oder absteigend erfolgen soll. Mit einem Klick auf „OK“ wird der ausgewählte Bereich nach den angegebenen Kriterien sortiert.

Zeilen und Spalten einer Tabelle vertauschen Um die Achsen einer Tabelle zu vertauschen, werden alle Zellen der Tabelle markiert und ausgeschnitten. Die Zelle A1 wird markiert und im Menü wird „Bearbeiten → Inhalte einfügen → Inhalte einfügen“ ausgewählt. Im Popup-Menü wird bei Optionen „Transponieren“ ausgewählt und der Vorgang mit „OK“ gestartet.

Leerzeilen aus einer Tabelle filtern und entfernen Um alle Leerzeilen aus einer Tabelle zu entfernen, wird ein Filter genutzt. Hierzu wird die gesamte Tabelle markiert und im Menü „Daten → Autofilter“ ausgewählt. Das Nutzen der ersten Zeile als Spaltenköpfe wird bestätigt und somit der Autofilter angewendet. Nun kann die Spaltenüberschrift einer Spalte angeklickt, die für alle nichtleeren Zeilen einen Wert enthält. In den Filtervorschlägen wird „Leer“ ausgewählt. Dadurch werden alle nichtleeren Zeilen ausgeblendet. Danach können alle Zeilen außer der ersten markiert und gelöscht werden. Dann wird der Autofilter aufgehoben und somit die nichtleeren Zeilen wieder eingeblendet, wodurch die Tabelle keine leeren Zeilen mehr enthält.

Datums- und Zeitangaben umformatieren Um Datums- und/oder Zeitangaben in ein anderes Format zu überführen, werden alle umzuformatierenden Zellen markiert. Nach einem Rechtsklick auf die markierten Zellen kann im Menü „Zellen formatieren“ ausgewählt werden. Im Popup-Menü wird der Reiter „Zahlen“, und dort die Kategorie „Datum“ oder „Uhrzeit“ ausgewählt. Daneben wird das aktuelle Format der Daten angezeigt. Hier wird stattdessen das gewünschte Format ausgewählt oder unten manuell der gewünschte Format-Code eingegeben. Mit einem Klick auf „OK“ werden alle markierten Zellen umformatiert. Wenn eine markierte Zelle keinen Wert in einem gültigen Datums- bzw. Zeitformat enthält, wird diese Zelle nicht umformatiert. Beim Importieren der Daten wird von MySQL dann eine Fehlermeldung ausgegeben.

Automatisches Generieren von MessungsIDs Um automatisch MessungsIDs zu generieren, kann eine Formel verwendet werden. Da es sich bei MessungsIDs um eine in Abhängigkeit von der Probennummer fortlaufende Nummer handelt, kann die MessungsID einer Zeile einfach berechnet werden, indem gezählt wird, wie oft die Probennummer der aktuellen Zeile zwischen der ersten und der aktuellen Zeile bereits vorkam. Da die erste Zeile in der Regel die Spaltenüberschriften enthält, fängt die Berechnung jedoch erst in der zweiten Zeile an. Hierzu wird die Formel =ZÄHLENWENN(A$2:A2, A2) in die zweite Zeile der Spalte `MessungsID` eingefügt, wobei A für die Spalte `Probennummer` steht und ggf. ausgetauscht werden muss, wenn `Probennummer` nicht die erste Spalte der Tabelle ist. Nun kann die Formel in die gesamte Spalte eingefügt werden, indem in die untere rechte Ecke der Zelle geklickt und bis zum Ende der Spalte gezogen wird, oder der im nächsten Paragraph beschriebene Vorgang zum Ausfüllen einer Spalte

mit dem gleichen Wert durchgeführt wird. Dabei muss darauf geachtet werden, dass das $ in der Formel vor der Zeilenangabe des ersten Parameters vorhanden ist. Dieses fixiert den Parameter als festen Wert 2, während alle anderen Parameter abhängig von der Position der Zelle sind, in der die Formel steht.

Spalten mit gleichen oder fortlaufenden Werten ausfüllen Um alle Zellen einer Spalte mit dem gleichen Wert oder einem fortlaufenden Wert zu füllen, wird zuerst der gewünschte (Start-)Wert in die erste Zeile der Spalte eingetragen. Dann wird der zu befüllende Bereich der Tabelle markiert. Um den Wert der ersten Zeile auf alle anderen Zeilen zu übertragen, wird im Menü „Tabelle → Zellen ausfüllen → Nach unten“ ausgewählt. Um den Startwert in der ersten Zeile in den weiteren Zeilen fortlaufen zu lassen, wird im Menü „Tabelle → Zellen ausfüllen → Mit Reihen“ ausgewählt. Daraufhin öffnet sich ein Popup-Menü, in dem die Richtung „Unten“ und der Reihentyp „Automatisch füllen“ ausgewählt werden. Mit einem Klick auf „OK“ wird die Spalte mit fortlaufenden Werten ausgefüllt.

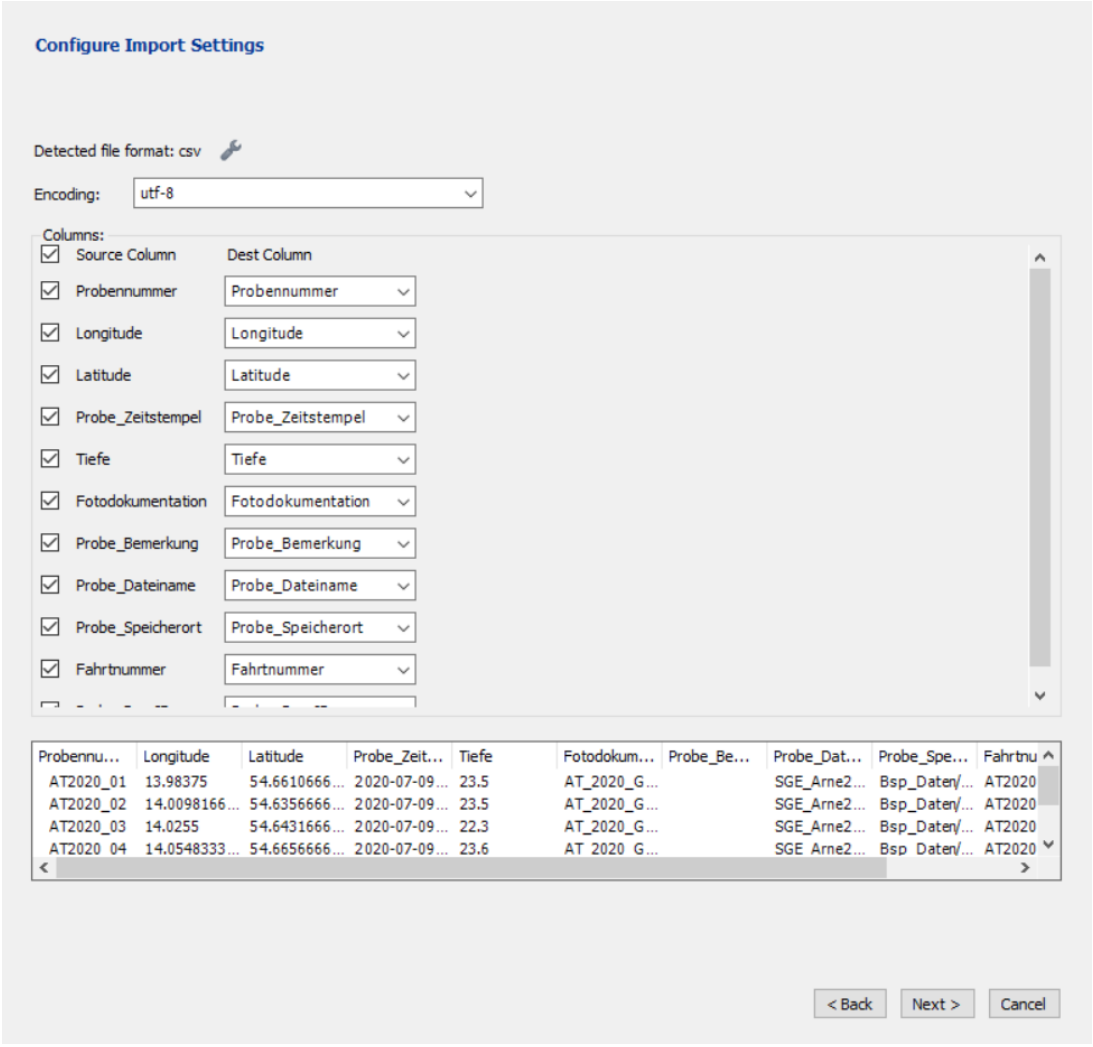

Abbildung 5.1: Spaltenzuordnung und Vorschau der AT2020-Daten im Table Data Import Wizard

Nach den Änderungen sollte für jede Relation eine eigene Datei existieren, welche nur die für diese Relation relevanten Spalten enthält. Die erste Zeile der Datei sollte die Spaltennamen enthalten, wobei die Spaltennamen in der Datei den Attributnamen der Relation entsprechen sollten. Eine Zeile muss einem Tupel der Datenbank entsprechen, und eine Spalte einem Attribut. Alle Attributwerte müssen atomar und im korrekten Format für das Attribut der Relation sein. Alle zu importierenden Dateien müssen eine Spalte mit den Provenance-IDs der Tupel erhalten. Wenn alle nötigen Änderungen an der Datei vorge-

nommen wurden, muss sie im .csv-Format gespeichert werden. Dabei ist auf die Trennzeichen zwischen Spalten und für Zeichenketten zu achten. Da insbesondere in Kommentarspalten Kommata vorkommen können, sollte als Trennzeichen das Semikolon verwendet werden. Als Trennzeichen für Zeichenketten sind Anführungszeichen dem einzelnen Hochkomma vorzuziehen, da Hochkommas beispielsweise in den Mastersizer-Daten vorhanden sind. Wenn die Datei im richtigen Format abgespeichert wurde, kann sie importiert werden, indem in der MySQL Workbench die korrespondierende Relation ausgewählt wird und mit einem Rechtsklick der Table Data Import Wizard geöffnet wird. Die zu importierende Datei wird ausgewählt und die Einstellungen für Dateiformat, Trennzeichen und Codierung des Wizards sollten überprüft und ggf. an die Datei angeglichen werden. Ebenso sollte die Zuordnung der Spaltennamen von Datei und Relation überprüft werden. In Abbildung 5.1 ist dieser Schritt für den Import der AT2020-Daten in die Relation Probe zu sehen. Im oberen Teil des Fensters können die Einstellungen bearbeitet werden, darunter kann die Zuweisung zwischen Spalten der Datei und Spalten der Relation bearbeitet werden, und unten ist eine Vorschau zu sehen, wie die Daten mit den aktuellen Einstellungen in die Datenbank eingefügt werden würden. Sind alle Einstellungen korrekt, kann der Import gestartet werden.

In den folgenden Abschnitten wird der genaue Prozess zur Import-Vorbereitung der Beispieldaten erläutert, die vom IOW für diese Arbeit zur Verfügung gestellt wurden. Jeder der folgenden Abschnitte behandelt die Dateien, die im jeweils gleichnamigen Abschnitt in Kapitel 4 vorgestellt wurden.

5.2.1 Altdaten

Um die Altdaten zu importieren, muss nur die Datei `all_cum.txt` modifiziert werden. Zunächst sollten die leeren Zeilen zwischen den Tupeln gelöscht werden. Als nächstes muss eine neue Spalte `Probennummer` eingefügt werden, da die Spalte `idx_map` einige doppelte Werte enthält und daher nicht als Probennummer und somit Primärschlüssel der Relation genutzt werden kann. Nachdem alle Leerzeilen entfernt wurden, kann hier einfach eine fortlaufende Nummer für alle Zeilen eingefügt werden. Außerdem wird eine Spalte `MessungsID` eingefügt, deren Wert für alle Tupel 1 ist. Zudem kann die Spalte `n` gelöscht werden. Optional können Umbenennungen der Spaltennamen vorgenommen werden, um das Importieren zu vereinfachen: `lon_wgs84` → `Longitude` und `lat_wgs84` → `Latitude`.

Da für die Altdaten keine Informationen über die Fahrt bekannt sind, werden keine Datensätze in die Relation FAHRT eingefügt. Stattdessen wird als erstes die Relation PROBE behandelt. Hierfür werden aus der vorbereiteten Datei die Spalten `Probennummer, Longitude` und `Latitude` in eine neue Datei kopiert. Es werden zusätzliche Spalten `Probe_Dateiname, Probe_Speicherort` und `Probe_ProvID` eingefügt. In die Spalte `Probe_Dateiname` wird für jedes Tupel „`all_cum.txt`“ und in die Spalte `Probe_Speicherort` der Speicherpfad, an dem die Datei gespeichert ist, eingefügt. In die Spalte `Probe_ProvID` wird für jedes Tupel eine Provenance-ID im Format „ProbeX“ mit X als fortlaufender Nummer eingefügt.

Als nächstes können für die Relation ALTDATEN die Spalten `Probennummer, idx_map, stat` und `quelle` in eine eigene Datei kopiert werden. Es wird eine Spalte `Altdaten_ProvID` hinzugefügt und für jedes Tupel eine Provenance-ID im Format „AltdatenX“ eingefügt.

Für die Relation MESSUNG werden die Spalten `Probennummer` und `MessungsID` in eine neue Datei kopiert. Es werden die Spalten `Einheit_Wert, Einheit_Korngroesse, Messung_Dateiname, Messung_-Speicherort` und `Messung_ProvID` hinzugefügt. Dabei wird für alle Tupel in der Spalte `Einheit_Wert` „kumulativ“, in der Spalte `Einheit_Korngroesse` „mikrometer“, in der Spalte `Messung_Dateiname` der Wert „`all_cum.txt`“, in der Spalte `Messung_Speicherort` der Speicherpfad der Datei und in der Spalte `Messung_ProvID` eine Provenance-ID im Format „MessungX“ eingetragen.

Für die Relation KLASSE wird eine Kopie der ersten angepassten Version von `all_cum.txt` erstellt und die Spalten `idx_map, stat, Longitude, Latitude` und `quelle` gelöscht, sodass nur die Spalten `Probennummer, MessungsID` und alle Spalten mit den Korngrößenklassen und dazugehörigen Messwerten bleiben. Da die Korngrößen und Messwerte jeweils in eigenen Spalten gespeichert sind, könnte die resultierende Datei theoretisch bereits mit dem Table Data Import Wizard in die Datenbank importiert werden. Da jedoch mehrere Wertepaare pro Tupel vorhanden sind, müsste der Wizard über 150 mal mit jeweils einer anderen Zuweisung von Spalten `d`X und `cum`X zu den Attributen `Korngroesse` und `Wert` der Relation. Dieser Prozess wäre sehr aufwändig und fehleranfällig und sollte daher vermieden werden. Stattdessen kann die Datei so umformatiert werden, dass pro Zeile nur ein Wertepaar vorkommt. Hierzu müssen die Spalte `Probennummer` und `Messungsnummer` jede Zeile so oft dupliziert werden, dass die Anzahl der Kopien einer Zeile der Anzahl der Wertepaare in der Zeile entspricht. Dann kann jedes Wertepaar in eine eigene Zeile verschoben werden. Die Spalte mit allen `d`X-Werten wird in `Korngroesse` umbenannt und die Spalte mit den `cum`X-Werten in `Wert`. In Anweisung 5.3 ist ein Algorithmus zum Umformatieren der Altdaten-Datei zum Import in die Relation KLASSE in Form von Pseudocode angegeben. Der Algorithmus ist allgemein für Dateistrukturen geeignet, die abwechselnd eine Spalte mit Korngrößen und eine Spalte mit den zugehörigen Messwerten der Korngröße der vorherigen Spalte enthalten.

5.2.2 AT2020

Um die AT2020-Daten zu importieren, müssen zwei Dateien modifiziert werden. Zuerst sollten die Metadaten in der Datei `SGE_Arne2020.xlsx` betrachtet werden, da hier die meisten Informationen über die Fahrt und die Proben vorhanden sind. Diese Daten müssen bereits in der Datenbank vorhanden sein, bevor Datensätze zu den Messungen bzw. Klassen eingefügt werden, da sonst Fremdschlüsselbedingungen verletzt werden und die Daten nicht importiert werden können. Aus dem Reiter `CruiseInfo` können die Informationen über die Fahrt AT2020 manuell mit einem INSERT INTO Statement in die Relation FAHRT eingefügt werden, da es sich nur um ein einziges Tupel handelt.

Um die Informationen über die Proben der Fahrt in die Datenbank einzufügen, kann der Reiter `Grabstation` in eine eigene Datei kopiert und an die Struktur der Relation PROBE angepasst werden. Hierfür werden alle Spalten außer `XNameGS, XXCoordinate, XYCoordinate, XSamplingDate, XDepth, XFoto` und `XRemark` gelöscht. Um den Zeitstempel der Probenentnahme als Datentyp DATETIME in die Relation einzufügen, müssen die Werte in der Spalte `XSamplingDate` dem Format YYYY-MM-DD hh:mm:ss entsprechen.

Optional können die Spaltennamen in der Datei in die Namen des entsprechenden Attributs der Relation PROBE umbenannt werden, um die Nutzung des Table Data Import Wizards zu vereinfachen. Sollen die Spalten umbenannt werden, sind folgende Umbenennungen nötig: `XNameGS` → `Probennnummer`, `XXCoordinate` → `Longitude`, `XYCoordinate` → `Latitude`, `XSamplingDate` → `Probe_Zeitstempel`, `XDepth` → `Tiefe`, `XFoto` → `Fotodokumentation` und `XRemark` → `Probe_Bemerkung`.

Des Weiteren müssen einige Spalten hinzugefügt werden. Dies sind die Spalten `Probe_Dateiname, Probe_Speicherort, Fahrtnummer und Probe_ProvID`. In `Probe_Dateiname` und `Probe_Speicherort` wird für jedes Tupel der Dateiname der Ursprungsdatei „`SGE_Arne2020.xlsx`“ und der Speicherpfad der Datei eingefügt. In `Fahrtnummer` wird für jedes Tupel der Wert AT2020 eingetragen, und in `Probe_ProvID` wird für jedes Tupel eine eigene Provenance-ID in der Form „ProbeX“ eingetragen, wobei es sich bei X um eine fortlaufende Nummer handelt.

```
function formatierenFuerKlasseAltdaten(anzahlProben, ersteKorngroessenSpalte,
    letzteSpalte, spalteProbennummer, spalteMessungsID){

        spalteKorngroesse = ersteKorngroessenSpalte + 2
        FOR EACH spalteKorngroesse <= letzteSpalte, spalteKorngroesse = spalteKorngroesse
            +2{

                spalteWert = spalteKorngroesse +1
                //Spaltenüberschriften entfernen
                tabelle[1][spalteKorngroesse] = NULL
                tabelle[1][spalteWert] = NULL

                zeile = 2
                FOR EACH zeile <= anzahlProben, zeile = zeile + 1{

                        IF tabelle[zeile][spalteKorngroesse] NOT NULL{
                                neueZeile = zeile + anzahlProben

                                //Probennummer und MessungsID der akutellen Zeile in die
                                    neue Zeile kopieren
                                probennummer = tabelle[zeile][spalteProbennummer]
                                messungsID = tabelle[zeile][spalteMessungsID]
                                tabelle[neueZeile][spalteProbennummer] = probennummer
                                tabelle[neueZeile][spalteMessungsID] = messungsID

                                //Messwert aus der aktuellen Zeile und Spalte
                                    ausschneiden
                                wert = tabelle[zeile][spalteWert]
                                tabelle[zeile][spalteWert] = NULL

                                //Korngröße aus der aktuellen Zeile und nächsten Spalte
                                    ausschneiden
                                korngroesse = tabelle[zeile][spalteKorngroesse]
                                tabelle[zeile][spalteKorngroesse] = NULL

                                //Messwert und Korngröße in die neue Zeile einfügen
                                tabelle[neueZeile][ersteKorngroessenSpalte] = korngroesse
                                tabelle[neueZeile][ersteKorngroessenSpalte + 1] = wert
                        }
                }
        }

        //Spaltenüberschriften für die Wertespalte und Korngrößenspalte einfügen
        tabelle[1][ersteKorngroessenSpalte] = 'Korngroesse'
        tabelle[1][ersteKorngroessenSpalte + 1] = 'Wert'
}
```

Anweisung 5.3: Pseudocode, Funktion zum Umformatieren einer Altdaten-Tabelle zum Import in die Relation Klasse

Als nächstes kann die Datei `AT2020_Grainsize - Kopie.xlsx` für die Relationen MESSUNG, SIEBEN und KLASSE kopiert und vorbereitet werden. Da alle Messungen in der Datei sowohl in Gramm als auch umgerechnet in Prozent aufgeführt werden, sollten zuerst die Zeilen 24 bis 38 gelöscht werden, sodass nur noch die Originalwerte in Gramm vorhanden sind. Außerdem können die Zeilen 1 bis 4 und Spalte A gelöscht werden, da diese Metainformationen enthalten, die in dieser Form den Import stören. Als nächstes wird die Spalte `Probe` in `Probennummer` umbenannt und eine zusätzliche Spalte `MessungsID` eingefügt und für jede Zeile eine MessungsID generiert.

Für die Relation MESSUNG werden die Spalten `Bemerkung, Messmethode, Probennummer` und `MessungsID` in eine eigene Datei kopiert und die Spalte `Bemerkung` in `Messung_Bemerkung` umbenannt. Es werden außerdem zusätzliche Spalten eingefügt: `Einheit_Korngroesse, Einheit_Wert, Messung_Dateiname, Messung_Speicherort` und `Messung_ProvID`. Diese Spalten müssen manuell mit Werten befüllt werden. In diesem Fall ist die Einheit der Korngröße „mikrometer“ und Einheit der Werte „gramm.“ In `Messung_Dateiname` wird für alle Tupel „AT2020_Grainsize - Kopie.xlsx“ und in `Messung_Speicherort` der Speicherpfad der Datei eingetragen. Die ProvID wird wie bereits beschrieben im Format „MessungX“ mit X als fortlaufender Nummer eingetragen.

Für die Relation SIEBEN werden die Spalten `Probennummer, MessungsID, gesplittet bei 2 mm, <63 µm abgeschlämmt, gesamt Gewicht trocken, trocken nach Schlämmen, Siebverlust, Summe aller Klassen` und `Muschelschalen` in eine eigene Datei kopiert. In der Spalte `gesplittet bei 2mm` müssen alle „ja“-Werte durch „2 mm“ ersetzt werden. In der Spalte `<63 µm abgeschlämmt` müssen alle „ja“-Werte durch „63 mikrometer“ ersetzt werden. Optional können folgende Umbennenungen der Spalten vorgenommen werden, um das Importieren zu vereinfachen: `gesplittet bei 2 mm` → `gesplittet_bei`, `<63 µm abgeschlämmt` → `abgeschlaemmt_bei, gesamt Gewicht trocken` → `Gesamtgewicht, trocken nach Schlämmen` → `Trockengewicht_nach_Schlämmung` und `Summe aller Klassen` → `Summe_aller_Klassen`. Außerdem muss die Spalte `Sieben_ProvID` hinzugefügt und im Format „SiebenX“ mit X als fortlaufender Nummer für alle Tupel eingetragen werden.

Als Letztes werden für die Relation KLASSE die Spalten `Probennummer, MessungsID` und alle Spalten rechts davon, welche die Korngrößenmesswerte enthalten, in eine neue Datei kopiert. Danach muss eine umfangreiche Umformatierung der Datei erfolgen. Es ist möglich, diese Umformatierung manuell vorzunehmen, jedoch ist dieser Prozess sehr aufwändig und sollte wenn möglich automatisiert werden. Ein Lösungsansatz zum automatischen Umformatieren in Form von Pseudocode ist in Anweisung 5.4 zu sehen. Dieser Algorithmus kann auf Dateistrukturen angewendet werden, bei denen alle Messwerte einer Korngröße in einer Spalte stehen, wobei die jeweilige Korngröße in der Spaltenüberschrift steht. Die Relation KLASSE speichert Tupel, die aus jeweils einer Probennummer, MessungsID, Korngrößenklasse und Messwert bestehen. In der Datei ist jedoch für jede Messung nur eine Zeile vorhanden, die alle Messwerte enthält. Die Korngrößenklasse ist jeweils Spaltenüberschrift des zugehörigen Messwerts. Das Format der Datei muss also in das Format der Relation überführt werden. Hierzu muss die Liste der Probennummern und MessungsIDs so oft untereinander kopiert werden wie die Anzahl der Korngrößenklassen in der Datei. Dann kann jeweils eine Spalte von Messwerten zu einem Satz Probennummern und MessungsIDs verschoben werden, wobei eine weitere Spalte benötigt wird, in welche die Korngrößenklasse aus der Spaltenüberschrift der Messwerte für alle betroffenen Zeilen eingetragen wird. Dieser Vorgang wird für alle Korngrößenklassen wiederholt, bis jede Zeile aus nur 4 Spalten besteht: `Probennummer, MessungsID, Wert, Korngroesse.` Zuletzt wird eine weitere Spalte `Klasse_ProvID` eingefügt, in der jedes Tupel eine ProvenanceID im Format „KlasseX“ erhält.

```
function formatierenFuerKlasse(anzahlProben, ersteWerteSpalte, letzteSpalte,
    spalteProbennummer, spalteMessungsID){

        spalte = ersteWerteSpalte + 1
        FOR EACH spalte <= letzteSpalte{

                //Korngröße der Spalte aus der Spaltenüberschrift ausschneiden
                korngroesse = tabelle[1][spalte]
                tabelle[1][spalte] = NULL

                zeile = 2
                FOR EACH zeile <= anzahlProben{

                        IF tabelle[zeile][spalte] NOT NULL{
                                neueZeile = zeile + anzahlProben

                                //Probennummer und MessungsID der akutellen Zeile in die
                                    neue Zeile kopieren
                                probennummer = tabelle[zeile][spalteProbennummer]
                                messungsID = tabelle[zeile][spalteMessungsID]
                                tabelle[neueZeile][spalteProbennummer] = probennummer
                                tabelle[neueZeile][spalteMessungsID] = messungsID

                                //Messwert aus der aktuellen Zeile und Spalte
                                    ausschneiden
                                wert = tabelle[zeile][spalte]
                                tabelle[zeile][spalte] = NULL

                                //Messwert und Korngröße in die neue Zeile einfügen
                                tabelle[neueZeile][ersteWerteSpalte] = wert
                                tabelle[neueZeile][ersteWerteSpalte + 1] = korngroesse
                        }
                }
        }

        //Korngröße aus der Überschrift der ersten Wertespalte ausschneiden
        korngroesse = tabelle[1][ersteWerteSpalte]

        //Korngröße in die ersten Zeilen einfügen, in denen anfangs noch die Messwerte
            der nächsten Korngröße standen
        zeile = 2
        FOR EACH zeile <= (anzahlProben + 1){
                tabelle[zeile][ersteWerteSpalte + 1] = korngroesse
        }

        //Spaltenüberschriften für die Wertespalte und Korngrößenspalte einfügen
        tabelle[1][ersteWerteSpalte] = 'Wert'
        tabelle[1][ersteWerteSpalte + 1] = 'Korngroesse'
}
```

Anweisung 5.4: Pseudocode, Funktion zum Umformatieren einer Tabelle zum Import in Relation Klasse

5.2.3 Atlas Projekt

In den zur Verfügung gestellten Daten zum Atlas Projekt sind keine Informationen über die Fahrt vorhanden, daher werden keine Tupel in die Relation FAHRT importiert. Stattdessen wird zuerst die Datei `PG_alle_Stationen_Sedimente.dbf` aus dem Ordner `Metadaten` betrachtet und zum Import in die Relation PROBE vorbereitet. Hierfür werden die Spalten `ID Name,C,254; Long,N,19,11` und `Lat,N,19,11` in eine eigene Datei kopiert und wie folgt umbenannt: `ID Name,C,254` → `Probennummer; Long,N,19,11` → `Longitude` und `Lat,N,19,11` → `Latitude`. Da manche Probennummern zusätzliche Informationen enthalten, wird eine Spalte `Probe_Bemerkung` eingefügt, in die diese Informationen verschoben werden können, sodass die Spalte `Probennummer` nur reine Probennummern enthält. Zusätzlich werden die Spalten `Probe_Dateiname, Probe_Speicherort` und `Probe_ProvID` erstellt. Dabei wird in `Probe_Dateiname` für jedes Tupel „`PG_alle_Stationen_Sedimente.dbf`“ und in `Probe_Speicherort` der Speicherpfad der Datei eingetragen. In der Spalte `Probe_ProvID` wird eine Provenance-ID im Format „ProbeX“ eingetragen.

Als Nächstes wird eine Kopie der Datei `Alle_Sedimentproben_PG.xlsx` erstellt und für den Import in die Relationen MESSUNG und KLASSE vorbereitet. Da die Datei mit umgekehrten Achsen formatiert ist, d.h. Tupel sind Spalten und Attribute sind Zeilen, müssen zuerst die Spalten und Zeilen vertauscht werden. Die Spalte `Sample Identity` wird in `Probennummer` umbenannt und eine neue Spalte `MessungsID` eingefügt. Da in dieser Datei keine Probe mehrfach vorkommt, kann in `MessungsID` für alle Zeilen der Wert 1 eingefügt werden.

Für die Relation MESSUNG werden die Spalten `Probennummer` und `MessungsID` kopiert und in eine neue Datei eingefügt. Es werden die Spalten `Messmethode, Einheit_Wert, Einheit_Korngroesse, Messung_Dateiname, Messung_Speicherort` und `Messung_ProvID` eingefügt. Bis auf `Messung_ProvID` wird in alle der neuen Spalten jeweils der gleiche Wert in alle Zeilen eingefügt. In der Spalte `Messmethode` erhalten alle Zeilen den Wert „Sieben“, in `Einheit_Wert` den Wert „prozent“, in `Einheit_Korngroesse` den Wert „mikrometer“, in `Messung_Dateiname` den Wert „`Alle_Sedimentproben_PG.xlsx`“ und in `Messung_Speicherort` den Speicherpfad der Datei. In der Spalte `Messung_ProvID` erhält jede Zeile eine Provenance-ID in der Form „MessungX“.

Obwohl es sich um gesiebte Proben handelt, werden keine Daten des Atlas Projektes in die Relation SIEBEN importiert. Dies liegt daran, dass im Gegensatz zu den Siebdaten aus Abschnitt 5.2.2 keine weiteren Informationen zum Siebvorgang gespeichert wurden. Die Daten enthalten zwar eine Spalte `Initial Sample Weight`, in der theoretisch das Gesamtgewicht der Probe enthalten sein sollte. Stattdessen enthält die Spalte jedoch für alle Proben den Wert 100, da die Messwerte der Korngrößenanalyse in Prozent angegeben werden und somit die Summe aller Messwerte 100% entspricht.

Für die Relation KLASSE wird eine Kopie der vorbereiteten Datei erstellt und dort die Spalten `Analyst, Date` und `Initial Sample Weight` entfernt, sodass nur die Spalten `Probennummer, MessungsID` und die Spalten mit den Messwerten bleiben. Die resultierende Datei muss jedoch noch umformatiert werden, damit jedes Wertepaar in einer eigenen Zeile steht. Dazu werden die Einträge der Spalten `Probennummer` und `MessungsID` untereinander dupliziert, sodass genauso viele Kopien der Zeilen existieren wie die Anzahl der Messwerte pro Zeile. Dann können die Spalten mit Messwerten jeweils in ihre eigene Zeilen verschoben werden, wobei die Spaltenüberschrift jeder Spalte von Messwerten in eine eigene Spalte verschoben wird. Auch für die Daten des Atlas Projekts kann der Algorithmus aus Anweisung 5.4 angewendet werden.

5.2.4 Mastersizer

In den Beispieldateien im Ordner `Mastersizer` befinden sich Messungen von den gleichen Proben wie im Abschnitt 5.2.2. Wenn diese schon importiert wurden, befinden sich schon alle benötigten Datensätze in den Relationen FAHRT und PROBE. Wurden diese noch nicht importiert, müssen zuerst die ersten Schritte des Abschnitts 5.2.2 durchgeführt werden. Als nächstes kann dann eine Kopie der Datei `AT2020_wet.txt` angefertigt und zum Import in die Relationen MESSUNG, MASTERSIZER und KLASSE vorbereitet werden. Hierfür werden zunächst drei neue Spalten `Probennummer, MessungsID` und `Messung_Bemerkung` eingefügt. Die Spalte `Sample Name` kann in ihrer originalen Form nicht einfach als Probennummer verwendet werden, da die Probennummern Zusatzinformationen enthalten, die automatisch von der Mastersizer-Software erstellt werden. In der Datenbank müssen jedoch die reinen Probennummern verwendet werden. Daher wird die zusätzliche Spalte `Probennummer` verwendet, um die reinen Probennummern einzutragen. Um die Zusatzinformationen aus `Sample Name` auch in der Datenbank zu speichern, wird die Spalte `Messung_Bemerkung` verwendet. So wird beispielsweise der Sample Name „Average of 'AT2020_23_1'“ aufgeteilt in die Probennummer „AT2020_23“ und die Bemerkung „Durchschnitt von Messungen 1-5“. Hierfür werden zuerst die Probennummern für jede Zeile bestimmt. Danach müssen die MessungsIDs jeder Zeile eingetragen werden. Hierfür wird in der Spalte `MessungsID` eine fortlaufende Nummer in Abhängigkeit von der Probennummer vergeben. Wenn die Probennummern und MessungsIDs aller Zeilen eingetragen wurden, können die Zusatzinformationen aus der Spalte `Sample Name` in die Spalte `Messung_Bemerkung` übertragen werden, wobei sie so umformuliert werden, dass bspw. für Durchschnittswerte die MessungsIDs der zugrundeliegenden Messungen angegeben werden. Danach kann die Spalte `Sample Name` gelöscht werden.

Für die Relation MESSUNG werden die Spalten `Probennummer, MessungsID, Messung_Bemerkung, Measurement Date Time` und `Operator Name` in eine eigene Datei kopiert. Die Spalten `Measurement Date Time` und `Operator Name` werden in `Messung_Zeitstempel` und `Analyst` umbenannt. Außerdem müssen die Werte in `Messung_Zeitstempel` umformatiert werden, da die Datenbank nur Werte im Format YYYY-MM-DD hh:mm:ss akzeptiert. Als nächstes werden die zusätzlichen Spalten `Messmethode, Einheit_Wert, Einheit_Korngroesse, Messung_Dateiname, Messung_Speicherort` und `Messung_ProvID` hinzugefügt. In die Spalte `Messmethode` wird für jedes Tupel „mastersizer_wet“ eingefügt, in `Einheit_Wert` „prozent“, in `Einheit_Korngroesse` „mikrometer“, in `Messung_Dateiname` „`AT2020_wet.txt`“, und in `Messung_Speicherort` der Speicherpfad der Datei. In der Spalte `Messung_ProvID` wird für jedes Tupel eine ProvenanceID im Format „MessungX“ eingefügt.

Für die Relation MASTERSIZER werden die Spalten `Probennummer, MessungsID, Dx(10), Dx(50)` und `Dx(90)` kopiert und in eine neue Datei eingefügt. Außerdem wird eine neue Spalte `Mastersizer_ProvID` eingefügt, in der jedes Tupel eine Provenance-ID im Format „MastersizerX“ erhält.

Für die Relation KLASSE wird eine Kopie der vorbereiteten Datei angefertigt und die Spalten `Record Number, Sample Name, Messung_Bemerkung, Measurement Date Time, Dx(10), Dx(50), Dx(90)` und `Operator Name` gelöscht, sodass nur die Spalten `Probennummer, MessungsID` und die Messwerte bleiben. Die Datei muss nun so umformatiert werden, dass in jeder Zeile nur ein Messwert vorhanden ist, und die Korngrößenklassen eine eigene Spalte erhalten, statt nur als Spaltenname der Messwerte zu existieren. Hierzu wird die Liste der Probennummern und MessungsIDs so oft dupliziert wie die Anzahl der Korngrößenklassen in der Datei. Dann können die Messwerte jeder Spalte so verschoben werden, dass jeder Zeile einen Messwert enthält. Außerdem muss beim Verschieben einer Spalte von Messwerten die Korngrößenklasse aus dem Spaltennamen ausgeschnitten und in einer eigenen Spalte in jedes Tupel aus der ursprünglichen Spalte eingefügt werden. Die Spalte mit allen Messwerten wird `Wert` und die Spalte mit allen Korngrößenklassen wird in `Korngroesse` umbenannt. Außerdem wird eine Spalte `Klasse_ProvID`

hinzugefügt, in der jedes Tupel eine Provenance-ID im Format „KlasseX“ erhält. Auch für die Mastersizer-Daten kann der Algorithmus aus Anweisung 5.4 angewendet werden.

5.2.5 Datenbankanfragen

Nach der Implementierung der Datenbank und dem Import der vorhandenen Daten können nun Anfragen auf die Datenbank erstellt werden. In diesem Abschnitt werden exemplarisch zwei Anfragen vorgestellt, die im Rahmen der Datenanalyse genutzt werden können. Weitere Datenbankanfragen und ihre Ergebnisse sind im Anhang der Arbeit zu finden.

```
SELECT * FROM klasse
WHERE Probennummer = 'AT2020_01' AND MessungsID = 1;
```

Anweisung 5.5: Anfrage aller Messwerte und Korngrößen der ersten Messung der Probe AT2020_01

Probennummer	MessungsID	Korngroesse	Wert	Klasse_ProvID
AT2020_01	1	1	0.14	Klasse577
AT2020_01	1	63	0.58	Klasse561
AT2020_01	1	75	0.75	Klasse545
AT2020_01	1	90	0.9	Klasse529
AT2020_01	1	106	1.1	Klasse513
AT2020_01	1	125	1.19	Klasse497
AT2020_01	1	150	1.43	Klasse481
AT2020_01	1	180	1.29	Klasse465
AT2020_01	1	212	1.61	Klasse449

Abbildung 5.2: Resultat von Anweisung 5.5

Anweisung 5.5 ist eine Datenbankanfrage, die alle Tupel der Relation KLASSE ausgibt, die zu einer bestimmten Messung gehören. Hier wurde die Messung mit der Probennummer „AT2020_01“ und MessungsID „1“ ausgewählt, das Ergebnis der Anfrage ist in Abbildung 5.2 zu sehen. Es enthält alle gemessenen Korngrößen und die dazugehörigen Messwerte der Klasse, sowie die Probennummer, MessungsID und Provenance-ID der Tupel. Andere Messungen können ausgewählt werden, indem die entsprechenden Werte in der WHERE-Klausel der Anfrage geändert werden.

```
SELECT * FROM probe
WHERE Probe_Zeitstempel BETWEEN '2020-01-01' AND '2020-12-31';
```

Anweisung 5.6: Anfrage aller Informationen über alle Proben, die zwischen dem 1. Januar 2020 und dem 31. Dezember 2020 entnommen wurden

Probennummer	Probe_Zeitstempel	Tiefe	Latitude	Longitude	Probe_Quelle	Bordbeschreibung	Fotodokumentation	Probe_Bemerkung
AT2020_01	2020-07-09 07:03:00	23.5	54.6611	13.9838	NULL	NULL	AT_2020_Greifer_RBK_01.JPG	
AT2020_02	2020-07-09 07:34:00	23.5	54.6357	14.0098	NULL	NULL	AT_2020_Greifer_RBK_02.JPG	
AT2020_03	2020-07-09 07:53:00	22.3	54.6432	14.0255	NULL	NULL	AT_2020_Greifer_RBK_03.JPG	
AT2020_04	2020-07-09 08:18:00	23.6	54.6657	14.0548	NULL	NULL	AT_2020_Greifer_RBK_04.JPG	
AT2020_05	2020-07-09 08:44:00	24.5	54.702	14.0545	NULL	NULL	AT_2020_Greifer_RBK_05.JPG	
AT2020_06	2020-07-09 09:00:00	24.6	54.713	14.0627	NULL	NULL	AT_2020_Greifer_RBK_06.JPG	
AT2020_07	2020-07-09 09:25:00	23.5	54.6977	14.0117	NULL	NULL	AT_2020_Greifer_RBK_07.JPG	
AT2020_08	2020-07-09 10:02:00	25.4	54.6882	13.9768	NULL	NULL	AT_2020_Greifer_RBK_08.JPG	grab sampler not c

Abbildung 5.3: Resultat von Anweisung 5.6

Anweisung 5.6 ist eine Datenbankanfrage, die alle Informationen zu allen Proben ausgibt, die in einem bestimmten Zeitraum entnommen wurden. Hier wurde als Zeitraum der 1. Januar 2020 bis zum 31.

Dezember 2020 ausgewählt. Das Ergebnis der Anfrage ist in Abbildung 5.3 zu sehen. Der gewünschte Zeitraum kann ausgewählt werden, indem die Werte in der WHERE-Klausel der Anfrage geändert werden.

Somit wurde die Implementierung der Datenbank abgeschlossen. Mit Ausnahme der Informationen der Relation KLASSE für die Altdaten- und Mastersizer-Dateien wurden alle vorhandenen Daten auf den Import in die Datenbank vorbereitet und importiert. Die relevanten Spalten dieser beiden Dateien konnten zum Zeitpunkt der Abgabe der Arbeit nicht auf den Import vorbereitet werden, da die manuelle Umformatierung zu aufwändig ist und der Algorithmus zur automatischen Umformatierung bisher nur in Form von Pseudocode vorhanden ist. Die AT2020- und Atlas-Projekt-Daten wurden manuell für den Import in die Relation KLASSE umformatiert und importiert. Durch die Datenbankanfragen wurden mögliche Anwendungen der Datenbank aufgezeigt.

6 Fazit

Dieses Kapitel bildet den Abschluss der Bachelorarbeit. Es werden zunächst allgemeine Erkenntnisse über das Forschungsdatenmanagement zusammengefasst. Dann erfolgt eine Zusammenfassung und Bewertung der Arbeit. Zuletzt folgt ein Ausblick auf mögliche weiterführende Themen, die im Rahmen dieser Arbeit nicht behandelt werden konnten.

6.1 Gewonnene Erkenntnisse

In diesem Abschnitt werden Erkenntnisse über das Forschungsdatenmanagement allgemein und am IOW behandelt, die im Laufe der Arbeit gewonnen wurden. Die wohl wichtigste dieser Erkenntnisse ist, dass im Rahmen der Datenerhebung so viele Metadaten wie möglich in der richtigen Form gespeichert werden sollten. Das Speichern zusätzlicher Metadaten zum Zeitpunkt der Datenerhebung bedarf nur geringem Aufwand, während das nachträgliche Ermitteln dieser Informationen oft aufwändig und fehleranfällig, wenn nicht sogar unmöglich ist.

In einigen der Datenstrukturen, die in dieser Arbeit betrachtet wurden, sind wichtige Metadaten nicht gespeichert. Dazu gehören beispielsweise Zeitstempel der Datenerhebung und der Name der Person, die die Datenerhebung durchgeführt hat. Im Falle der Beispieldaten des Atlas Projekts (siehe Kapitel 4.1.3) sind diese Informationen sogar als Spalten in der Datei vorhanden, wurden jedoch nicht ausgefüllt. Andere Metadaten können ggf. nachträglich ermittelt werden, indem beispielsweise die Daten genau überprüft werden (z.B. ist die Einheit der Messwerte höchstwahrscheinlich Prozent, wenn für alle Messungen in der Datei die Summe ihrer Messwerte exakt 100 ergibt). Da das nachträgliche Ermitteln dieser Informationen jedoch aufwändig, fehleranfällig und nicht immer möglich ist, sollten sie in Zukunft immer explizit in den Datensätzen gespeichert werden. Dies betrifft zum Beispiel verwendete Mess- und Aufbereitungsmethoden oder Einheiten.

Ein anderes Problem ist die korrekte Verwendung von Spalten, da beispielsweise die Spalte `Initial Sample Weight` in der Beispieldatei für das Atlas Projekt (zu sehen in Abbildung 4.6 in Kapitel 4.1.3) nicht wie vorgesehen das Gesamtgewicht der Probe beinhaltet, sondern den Wert 100 als Summe aller Klassen, die in Prozent angegeben sind. Ebenso sind in den Mastersizer-Daten (siehe Abbildung 4.8 in Kapitel 4.1.4) zusätzliche Informationen in der Spalte für Probennummern enthalten, obwohl diese auch in einer separaten Spalte für Bemerkungen gespeichert werden könnten. Wenn solche zweckentfremdeten Spalten bei der Datenintegration oder -analyse nicht auffallen und behoben werden, können sie zu Fehlern führen. So würde MySQL beispielsweise aufgrund verletzter Fremdschlüsselbedingungen eine Fehlermeldung liefern und den Import der Mastersizer-Daten nicht durchführen, wenn die weiteren Informationen der Spalte `Probennummer` nicht vorher entfernt und in eine andere Spalte verschoben wurden.

Des Weiteren wäre es von Vorteil, für die verschiedenen Messverfahren feste Schemata vorzuschreiben, in denen die Daten gespeichert werden müssen. Somit könnte für jedes der vorgeschriebenen Schemata ein Skript geschrieben werden, welches die Datenintegration automatisch durchführt.

Manche dieser Probleme werden erst kritisch, wenn Daten aus den Dateien in Datenbanken importiert werden sollen. So stellen beispielsweise Zusatzinformationen in der Spalte `Probennummer` kein Problem für die manuelle Datenanalyse dar, da Wissenschaftler leicht zwischen den Zusatzinformationen und der eigentlichen Probennummer differenzieren können. Für den Import in eine Datenbank müssen jedoch alle Informationen unmissverständlich, atomar und unter der korrekten Bezeichnung gespeichert sein. Die Umstellung auf solche Speicherstrukturen kann umständlich sein. Da die Datenhaltung in Datenbanken jedoch viele Vorteile wie Zentrale Speicherung, Zugriff mehrerer Nutzer, Vermeidung von Redundanzen, Konsistenz der Daten und vielfältige Anfragemöglichkeiten bieten, sollten alle Daten so gespeichert werden, dass sie möglichst einfach in Datenbanken importiert werden können. Dies verringert den nötigen Aufwand des Umformatierens, wenn sich in Zukunft für die Nutzung von Datenbanken entschieden wird.

6.2 Zusammenfassung

Das Ziel dieser Arbeit war der Entwurf und prototypische Implementierung einer geeigneten Datenbank zur Speicherung von Sedimentdaten des IOW, unter Einbindung von Data- und/oder Workflow Provenance. Hierzu wurden die einzelnen Dateien genau auf ihre Struktur und die darin enthaltenen Informationen analysiert und zu jeder Speicherform ein Entity-Relation-Modell erstellt. Die verschiedenen Modelle wurden verglichen und so zu einem Gesamtmodell kombiniert, dass alle Informationen der einzelnen Modelle auch in dem Gesamtmodell vertreten sind. Des Weiteren wurde in Absprache mit dem IOW erarbeitet, welche zusätzlichen Informationen und Metadaten gespeichert werden sollten, die nicht explizit in den Daten vorhanden sind. Danach wurde das Gesamtmodell um Tupel-IDs erweitert, um die Rückverfolgung von Tupeln innerhalb der Datenbank, und somit Data Provenance zu ermöglichen. Außerdem wurde das Modell um zusätzliche Entity-Typen erweitert, in denen weitere Metadaten gespeichert werden können, welche beispielsweise die Rückverfolgung von Datensätzen ermöglichen, die in bestimmten Projekten oder wissenschaftlichen Arbeiten verwendet wurden. Das fertige ER-Modell wurde auf ein Relationenmodell abgebildet und normalisiert. Auf Basis des normalisierten Relationenmodells wurde mit SQL-Statements eine relationale Datenbank in MySQL erstellt. Es wurden allgemeine Empfehlungen für die nötigen Modifikationen an Originaldateien entworfen, die zur Vorbereitung auf den Datenimport durchgeführt werden müssen. Dann wurden die Beispieldatensätze so modifiziert, dass sie in die Datenbank importiert werden konnten. Hierzu wurde auch ein Algorithmus in Form von Pseudocode entworfen, welcher die aufwändigste der nötigen Modifikationen, das Umformatieren der Korngrößenklassen und Messwerte, automatisiert. Zuletzt wurden die Beispieldatensätze in die erstellte Datenbank importiert.

Mit der Datenstrukturanalyse, dem Entwurf und der Implementierung der Datenbank wurde das Hauptziel der Arbeit erfüllt. Durch das Einfügen von global eindeutigen Tupel-IDs wurde die Datenbank auf die Integration von Data Provenance vorbereitet. Workflow Provenance wurde hingegen nicht umgesetzt. Stattdessen wurden Aspekte von Metadata Provenance in die Datenbank mit einbezogen, indem eine Reihe von zusätzlichen Metadaten in den Datenbankentwurf eingefügt und implementiert wurden.

6.3 Ausblick

Im Verlauf dieser Arbeit wurde der Fokus von Workflow Provenance auf Metadata Provenance verschoben. Die Umsetzung von Workflow Provenance für die hier behandelten Sedimentdaten ist jedoch weitere

Betrachtung wert. In der entworfenen Datenbank werden nur grundlegende Informationen über die untersuchten Proben, die Analyseergebnisse und zugehörige Metadaten gespeichert und die Option zur Verfügung gestellt, die Beziehung zwischen Projekten bzw. wissenschaftlichen Arbeiten und den dafür verwendeten Messungen zu speichern. Hierzu wird jedoch nur gespeichert, welche Messungen im Rahmen welches Projektes erhoben wurden. Mithilfe von Techniken der Workflow Provenance könnten weitere wichtige Informationen gespeichert werden wie Zwischenergebnisse und verwendete Analysetechniken. Dies würde die Rückverfolgbarkeit von Forschungsergebnissen signifikant erhöhen.

Des Weiteren wurde die Datenbank in dieser Arbeit durch den Einsatz von Tupel-IDs nur auf die Umsetzung von Data Provenance vorbereitet. Es fand keine Implementierung von Techniken zur Abfrage von Where-, Why- und How-Provenance statt. Where-Provenance kann ggf. durch das richtige Einbinden der ProvID-Spalten in der SELECT-Klausel von SQL-Anfragen erreicht werden, um die komplexeren Provenance-Abfragen zu ermöglichen müsste jedoch weitere Implementation erfolgen.

Aufgrund der Heterogenität der Datensätze, die in dieser Arbeit behandelt wurden, lässt sich der Datenimport nur schwer automatisieren. Auf Basis der hier entwickelten Datenbank könnte jedoch ein festes Speicherschema für jede Messmethode entworfen werden, welche als Exportvorlage für zukünftige Datenerhebungen genutzt werden kann. Wenn nur noch eine geringe Menge festgelegter Exportschemata genutzt werden, könnten Skripte zur automatischen Transformation der Dateien und Import in die Datenbank entwickelt werden.

7 Anhang

DDLs

```
CREATE TABLE fahrt (
Fahrtnummer VARCHAR(20) PRIMARY KEY,
Gebiet VARCHAR(100),
ChiefScientist VARCHAR(50),
IMO CHAR(7),
von DATE,
bis DATE
);

CREATE TABLE probe (
Probennummer VARCHAR(30),
Probe_Zeitstempel DATETIME,
Tiefe FLOAT,
Latitude FLOAT,
Longitude FLOAT,
Probe_Quelle VARCHAR(255),
Bordbeschreibung VARCHAR(150),
Fotodokumentation VARCHAR(150),
Probe_Bemerkung VARCHAR(500),
Fahrtnummer VARCHAR(20),
PRIMARY KEY(Probennummer),
FOREIGN KEY(Fahrtnummer) REFERENCES fahrt(Fahrtnummer)
);

CREATE TABLE altdaten(
Probennummer VARCHAR(30),
Idx_map VARCHAR(20),
stat varchar(20),
Altdaten_Quelle varchar(255),
PRIMARY KEY(Probennummer),
FOREIGN KEY(Probennummer) REFERENCES probe(Probennummer)
);

CREATE TABLE messung(
Probennummer VARCHAR(30),
MessungsID INT UNSIGNED NOT NULL,
Messmethode ENUM('sieben','mastersizer_wet','mastersizer_aero'),
Aufbereitung_Organik VARCHAR(20),
Aufbereitung_Carbonat VARCHAR(20),
Einheit_Wert ENUM('kumulativ','prozent','gramm','milligramm'),
Einheit_Korngroesse ENUM('millimeter','mikrometer'),
Messung_Zeitstempel DATETIME,
Analyst VARCHAR(50),
Messung_Bemerkung VARCHAR(255),
PRIMARY KEY(Probennummer, MessungsID),
FOREIGN KEY(Probennummer) REFERENCES probe(Probennummer)
);
```

```
CREATE TABLE sieben(
Probennummer VARCHAR(30),
MessungsID INT UNSIGNED,
Gesplittet_bei VARCHAR(10),
Abgeschlaemmt_bei VARCHAR(10),
Gesamtgewicht FLOAT,
Summe_aller_Klassen FLOAT,
Siebverlust FLOAT,
Muschelschalen FLOAT,
Trockengewicht_nach_Schlaemmung FLOAT,
PRIMARY KEY(Probennummer, MessungsID),
FOREIGN KEY(Probennummer, MessungsID) REFERENCES messung(Probennummer, MessungsID)
);

CREATE TABLE mastersizer(
Probennummer VARCHAR(30),
MessungsID INT UNSIGNED,
Dx10 DOUBLE,
Dx50 FLOAT,
Dx90 FLOAT,
PRIMARY KEY(Probennummer, MessungsID),
FOREIGN KEY(Probennummer, MessungsID) REFERENCES messung(Probennummer, MessungsID)
);

CREATE TABLE klasse(
Probennummer VARCHAR(30),
MessungsID INT UNSIGNED,
Korngroesse FLOAT,
Wert FLOAT,
PRIMARY KEY(Probennummer, MessungsID,Korngroesse),
FOREIGN KEY(Probennummer, MessungsID) REFERENCES messung(Probennummer, MessungsID)
);

CREATE TABLE projekt(
Projektname VARCHAR (50),
Art VARCHAR(30),
Ansprechpartner VARCHAR(50),
PRIMARY KEY (Projektname)
);

CREATE TABLE teil_von(
Probennummer VARCHAR(30),
MessungsID INT UNSIGNED,
Projektname VARCHAR(50),
PRIMARY KEY (Probennummer, MessungsID, Projektname),
FOREIGN KEY(Probennummer) REFERENCES messung(Probennummer),
FOREIGN KEY(Probennummer, MessungsID) REFERENCES messung(Probennummer, MessungsID)
);

CREATE TABLE projekt_doi(
Projektname VARCHAR(50),
DOI VARCHAR(255),
PRIMARY KEY(Projektname, DOI),
FOREIGN KEY(Projektname) REFERENCES projekt(Projektname)
);


ALTER TABLE probe ADD COLUMN Probe_Dateiname VARCHAR(100);
ALTER TABLE probe ADD COLUMN Probe_Speicherort VARCHAR(255);
ALTER TABLE messung ADD COLUMN Messung_Dateiname VARCHAR(100);
ALTER TABLE messung ADD COLUMN Messung_Speicherort VARCHAR(255);
```

```
ALTER TABLE fahrt ADD COLUMN Fahrt_ProvID VARCHAR(30) UNIQUE;
ALTER TABLE probe ADD COLUMN Probe_ProvID VARCHAR(30) UNIQUE;
ALTER TABLE altdaten ADD COLUMN Altdaten_ProvID varchar(30) UNIQUE;
ALTER TABLE messung ADD COLUMN Messung_ProvID VARCHAR(30) UNIQUE;
ALTER TABLE sieben ADD COLUMN Sieben_ProvID VARCHAR(30) UNIQUE;
ALTER TABLE mastersizer ADD COLUMN Mastersizer_ProvID VARCHAR(30) UNIQUE;
ALTER TABLE klasse ADD COLUMN Klasse_ProvID VARCHAR(30) UNIQUE;
ALTER TABLE projekt ADD COLUMN Projekt_ProvID VARCHAR(30) UNIQUE;
ALTER TABLE teil_von ADD COLUMN Teil_von_ProvID VARCHAR(30) UNIQUE;
ALTER TABLE projekt_doi ADD COLUMN Projekt_DOI_ProvID VARCHAR(30) UNIQUE;

CREATE TRIGGER setMessungsID
        BEFORE INSERT ON messung
        FOR EACH ROW
        SET New.MessungsID := IFNULL(
                (SELECT MAX(MessungsID)+1
                        FROM messung
                        WHERE Probennummer = New.Probennummer ), 1);
```

Datenbankanfragen

```
SELECT Probennummer, Longitude, Latitude FROM probe
WHERE Probennummer = 'AT2020_01';
```

Anweisung 7.1: Anfrage der Koordinaten der Probe AT2020_01

	Probennummer	Longitude	Latitude
▶	AT2020_01	13.9838	54.6611
*	NULL	NULL	NULL

Abbildung 7.1: Resultat von Anweisung 7.1

```
SELECT * FROM messung
WHERE Probennummer = 'AT2020_01';
```

Anweisung 7.2: Anfrage aller Informationen über alle Messungen der Probe AT2020_01

	Probennummer	MessungsID	Messmethode	Aufbereitung_Organik	Aufbereitung_Carbonat	Einheit_Wert	Einheit_Korngroesse	M
▶	AT2020_01	1	sieben	NULL	NULL	gramm	mikrometer	NUL
*	NULL	NULL	NULL	NULL	NULL	NULL	NULL	NUL

Abbildung 7.2: Resultat von Anweisung 7.2

```
SELECT * FROM klasse
WHERE Probennummer = 'AT2020_01' AND MessungsID = 1;
```

Anweisung 7.3: Anfrage aller Messwerte und Korngrößen der ersten Messung der Probe AT2020_01

Probennummer	MessungsID	Korngroesse	Wert	Klasse_ProvID
AT2020_01	1	1	0.14	Klasse577
AT2020_01	1	63	0.58	Klasse561
AT2020_01	1	75	0.75	Klasse545
AT2020_01	1	90	0.9	Klasse529
AT2020_01	1	106	1.1	Klasse513
AT2020_01	1	125	1.19	Klasse497
AT2020_01	1	150	1.43	Klasse481
AT2020_01	1	180	1.29	Klasse465
AT2020_01	1	212	1.61	Klasse449

Abbildung 7.3: Resultat von Anweisung 7.3

```
SELECT * FROM probe
WHERE Probe_Zeitstempel BETWEEN '2020-01-01' AND '2020-12-31';
```

Anweisung 7.4: Anfrage aller Informationen über alle Proben, die zwischen dem 1. Januar und dem 31. Dezember 2020 entnommen wurden

Probennummer	Probe_Zeitstempel	Tiefe	Latitude	Longitude	Probe_Quelle	Bordbeschreibung	Fotodokumentation	Probe_Bemerkung
AT2020_01	2020-07-09 07:03:00	23.5	54.6611	13.9838	NULL	NULL	AT_2020_Greifer_RBK_01.JPG	
AT2020_02	2020-07-09 07:34:00	23.5	54.6357	14.0098	NULL	NULL	AT_2020_Greifer_RBK_02.JPG	
AT2020_03	2020-07-09 07:53:00	22.3	54.6432	14.0255	NULL	NULL	AT_2020_Greifer_RBK_03.JPG	
AT2020_04	2020-07-09 08:18:00	23.6	54.6657	14.0548	NULL	NULL	AT_2020_Greifer_RBK_04.JPG	
AT2020_05	2020-07-09 08:44:00	24.5	54.702	14.0545	NULL	NULL	AT_2020_Greifer_RBK_05.JPG	
AT2020_06	2020-07-09 09:00:00	24.6	54.713	14.0627	NULL	NULL	AT_2020_Greifer_RBK_06.JPG	
AT2020_07	2020-07-09 09:25:00	23.5	54.6977	14.0117	NULL	NULL	AT_2020_Greifer_RBK_07.JPG	
AT2020_08	2020-07-09 10:02:00	25.4	54.6882	13.9768	NULL	NULL	AT_2020_Greifer_RBK_08.JPG	grab sampler not c

Abbildung 7.4: Resultat von Anweisung 7.4

```
SELECT * FROM probe
WHERE Latitude BETWEEN 54.5 AND 54.6
AND Longitude BETWEEN 13.9 AND 14;
```

Anweisung 7.5: Anfrage aller Informationen über alle Proben, die in dem Bereich zwischen 54.5° und 54.6° Breite und 13.9° und 14° Länge entnommen wurden

Probennummer	Probe_Zeitstempel	Tiefe	Latitude	Longitude	Probe_Quelle	Bordbeschreibung	Fotodokumentation	Probe_Bemerkung	Fahrtnummer	Prob
30340	NULL	NULL	54.5116	13.9362	NULL	NULL	NULL	NULL	NULL	all_c
30341	NULL	NULL	54.5489	13.9673	NULL	NULL	NULL	NULL	NULL	all_c
30342	NULL	NULL	54.5853	13.9914	NULL	NULL	NULL	NULL	NULL	all_c
30385	NULL	NULL	54.5732	13.9419	NULL	NULL	NULL	NULL	NULL	all_c
30413	NULL	NULL	54.5643	13.9697	NULL	NULL	NULL	NULL	NULL	all_c
30415	NULL	NULL	54.5804	13.9843	NULL	NULL	NULL	NULL	NULL	all_c
30416	NULL	NULL	54.5892	13.9572	NULL	NULL	NULL	NULL	NULL	all_c
30417	NULL	NULL	54.598	13.9302	NULL	NULL	NULL	NULL	NULL	all_c

Abbildung 7.5: Resultat von Anweisung 7.5

```
SELECT * FROM probe
WHERE Fahrtnummer = 'AT2020';
```

Anweisung 7.6: Anfrage aller Informationen über alle Proben, die auf der Fahrt AT2020 entnommen wurden

Probennummer	Probe_Zeitstempel	Tiefe	Latitude	Longitude	Probe_Quelle	Bordbeschreibung	Fotodokumentation	Probe_Bemerkung
AT2020_01	2020-07-09 07:03:00	23.5	54.6611	13.9838	NULL	NULL	AT_2020_Greifer_RBK_01.JPG	
AT2020_02	2020-07-09 07:34:00	23.5	54.6357	14.0098	NULL	NULL	AT_2020_Greifer_RBK_02.JPG	
AT2020_03	2020-07-09 07:53:00	22.3	54.6432	14.0255	NULL	NULL	AT_2020_Greifer_RBK_03.JPG	
AT2020_04	2020-07-09 08:18:00	23.6	54.6657	14.0548	NULL	NULL	AT_2020_Greifer_RBK_04.JPG	
AT2020_05	2020-07-09 08:44:00	24.5	54.702	14.0545	NULL	NULL	AT_2020_Greifer_RBK_05.JPG	
AT2020_06	2020-07-09 09:00:00	24.6	54.713	14.0627	NULL	NULL	AT_2020_Greifer_RBK_06.JPG	
AT2020_07	2020-07-09 09:25:00	23.5	54.6977	14.0117	NULL	NULL	AT_2020_Greifer_RBK_07.JPG	
AT2020_08	2020-07-09 10:02:00	25.4	54.6882	13.9768	NULL	NULL	AT_2020_Greifer_RBK_08.JPG	grab sampler not co

Abbildung 7.6: Resultat von Anweisung 7.6

Digitale Abgabe

Die digitale Abgabe dieser Bachelorarbeit enthält folgende Ordner:

Latex: Dieser Ordner enthält die digitale Version der Arbeit (`BA-kristi-bartelt-final.pdf`), alle LaTeX-Quelldateien sowie den Ordner **Bilder**, der alle in der Arbeit genutzten Grafiken enthält.

Literatur: Dieser Ordner enthält die verwendeten Literaturquellen als .pdf-Dateien. Verwendete Internetquellen befinden sich im Unterordner **Websites**. Alle Dateinamen entsprechen dem Zitierschlüssel im Literaturverzeichnis.

Implementierung: Dieser Ordner enthält den Quellcode dieser Arbeit. Die Datei `DDL.sql` enthält die SQL-Anweisungen zur Erstellung der Datenbank, die Datei `Datenimport_Vorbereitung.sql` enthält Anweisungen zum Einfügen einzelner Datensätze und Datenbankanfragen zur Ermittlung der Tupelanzahl jeder Relation, die benötigt werden, um die Provenance-IDs der nächsten einzufügenden Tupel zu bestimmen. Die Datei `Anfragen.sql` enthält die Datenbankanfragen aus den Kapiteln 5.2.5 und 7. Die Datei `Dump20220322.sql` enthält ein Backup der gesamten Datenbank zum Zeitpunkt der Abgabe. Die Dateien `Pseudocode.txt` und `Pseudocode_Altdaten` enthalten den Algorithmus zur Umformatierung der Dateien.

Bsp_Daten: Dieser Ordner enthält alle exemplarischen Daten, die vom IOW für diese Arbeit zur Verfügung gestellt wurden. Die Dateien sind auf die Unterordner **Altdaten, AT2020, Atlas Projekt, Mastersizer** und **Metadata** verteilt, die in der Arbeit beschrieben wurden. Außerdem enthalten diese Unterordner die .csv-Dateien, welche in Vorbereitung auf den Datenimport erstellt wurden.

Literaturverzeichnis

[AMJ+20] Tanja Auge, Erik Manthey, Susanne Jürgensmann, Susanne Feistel, and Andreas Heuer. Schema Evolution and Reproducibility of Long-term Hydrographic Data Sets at the IOW. In *LWDA, CEUR Workshop Proceedings*, volume 2738, pages 258–2, 2020.

[BKM+17] Ilvio Bruder, Meike Klettke, Mark Lukas Möller, Frank Meyer, Andreas Heuer, Susanne Jürgensmann, and Susanne Feistel. Daten wie Sand am Meer – Datenerhebung, -strukturierung, -management und Data Provenance für die Ostseeforschung. *Datenbank-Spektrum*, 17(2):183–196, jun 2017.

[CCT09] James Cheney, Laura Chiticariu, and Wang-Chiew Tan. Provenance in Databases: Why, How, and Where. *Foundations and Trends in Databases*, 1(4):379–474, 2009.

[Fen18] Wolfgang Fennel. Meereskunde in Warnemünde – Ausgangslage, Abwicklung und Neugründung in der Zeit von 1990 bis 1994. *Meereswissenschaftliche Berichte No 106 2018 - Marine Science Reports No 106 2018*, 2018.

[HDB17] Melanie Herschel, Ralf Diestelkämper, and Houssem Ben Lahmar. A survey on provenance: What for? What form? What from? *The VLDB Journal*, 26(6):881–906, October 2017.

[HSS18] Andreas Heuer, Kai-Uwe Sattler, and Gunter Saake. *Datenbanken - Konzepte und Sprachen*. MITP, April 2018.

[Leia] Leibniz-Institut für Ostseeforschung Warnemünde. Forschungsprogramm 2013-2023. `https://www.io-warnemuende.de/forschungsprogramm.html`. zuletzt abgerufen am 13.01.2022.

[Leib] Leibniz-Institut für Ostseeforschung Warnemünde. Kurzvorstellung & Lageplan. `https://www.io-warnemuende.de/kurzvorstellung-lageplan.html`. zuletzt aufgerufen am 13.01.2022.

[Man20] Erik Manthey. Beschreibung der Veränderungen von Schemata und Daten am IOW mit Schema-Evolutions-Operatorens. Bachelorarbeit, 2020.

[Mey16] Frank Meyer. Temporale Aspekte und Provenance-Anfragen im Umfeld des Forschungsdatenmanagements. Masterarbeit, Universität Rostock, 2016.

[Mö16] Mark Lukas Möller. Aufbau einer Forschungsdatenverwaltung für chemische und physikalische In-Situ-Daten aus der Ostsee. Bachelorarbeit, 2016.

[Ora] Oracle Corporation. Data Definition Statements - MySQL 8.0 Reference Manual. `https://dev.mysql.com/doc/refman/8.0/en/sql-data-definition-statements.html`. zuletzt aufgerufen am 21.03.2022.

Anfragenverzeichnis

Tabellenverzeichnis

Abbildungsverzeichnis

Selbstständigkeitserklärung

Hiermit erkläre ich, dass ich die vorliegende Masterarbeit selbstständig und nur unter Verwendung der angegebenen Literatur und Hilfsmittel angefertigt habe. Die aus fremden Quellen direkt oder indirekt übernommen Stellen sind als solche kenntlich gemacht. Die Arbeit wurde bisher in gleicher oder ähnlicher Form keiner anderen Prüfungsbehörde vorgelegt und auch nicht veröffentlicht.

Rostock, den 22.3.2022

Bartelt