

Otto-von-Guericke-Universität Magdeburg

Fakultät für Informatik



Bachelorarbeit

Vergleich revisionsssicherer Archivierungsmethoden im Kontext von RDBMS-Applikationen

Autor:

Danny Görner

26. September 2016

Betreuer:

Prof. Dr. rer. nat. habil. Gunter Saake

Institut für Technische und Betriebliche Informationssysteme

M. Sc. Sebastian Dorok

Institut für Technische und Betriebliche Informationssysteme

Görner, Danny:

*Vergleich revisionssicherer Archivierungsmethoden im Kontext von RDBMS-
Applikationen*

Bachelorarbeit, Otto-von-Guericke-Universität Magdeburg, 2016.

Inhaltsangabe

Viele Institutionen sind gesetzlich dazu verpflichtet ihre Daten zu archivieren. Dies betrifft auch den Verein für Gesundheit, Bewegung und Sport (VGBS). Das Ziel dieser Arbeit ist es ein Archivierungskonzept für die datenbankgestützte Verwaltungsanwendung des VGBS zu entwickeln. Die Herausforderung hierbei besteht darin aus verschiedenen Archivierungslösungen eine auszuwählen. Aus diesem Grund werden die gegebenen Anforderungen analysiert sowie ein qualitativer Vergleich der entsprechenden Lösungen bezüglich der Anforderungen durchgeführt. Die Analyse der Anforderungen erfolgt basierend auf Fachliteratur, dem rechtlichen Kontext sowie auf den Anforderungen des VGBS. Die Ergebnisse zeigen einen Widerspruch zwischen zwei Anforderungen: Der gesetzlich vorgeschriebenen Unveränderbarkeit archivierter Daten und der Forderung des VGBS diese verändern zu können. Existierende Ansätze können diesen Widerspruch nicht lösen. Deshalb ist es erforderlich diese konzeptuell anzupassen. Daraus resultierend werden die datenbankinterne sowie die datenbankexterne Methode vorgestellt. Ferner wird aufgezeigt, wie beide Archivierungsmethoden diesen Widerspruch konzeptionell lösen können. Der anschließende qualitative Vergleich beider Methoden zeigt, dass die datenbankinterne Methode die gegebenen Anforderungen am besten erfüllen kann. Somit wird diese im Kontext der Verwaltungsanwendung des VGBS zukünftig Anwendung finden.

Inhaltsverzeichnis

Abbildungsverzeichnis	ix
Tabellenverzeichnis	xii
1 Einführung	1
2 Grundlagen	5
2.1 Analytic Hierarchy Process	5
2.1.1 Bewertungsskala	6
2.1.2 Ablauf	7
2.1.2.1 Definition des Entscheidungsproblems	7
2.1.2.2 Konstruktion der Hierarchie	7
2.1.2.3 Vergleich und Bewertung	7
2.1.2.4 Gewichtung und Lösung	9
2.2 Temporale Datenhaltung	11
2.2.1 Zentrale Konzepte	11
2.2.1.1 Granularität	11
2.2.1.2 Chronon	11
2.2.1.3 Zeitpunkt	11
2.2.1.4 Zeitintervall	12
2.2.1.5 Temporales Element	12
2.2.1.6 Zeitstempelung	12
2.2.1.7 Schnappschuss	12
2.2.1.8 Benutzerdefinierte Zeit	12
2.2.1.9 Gültigkeitszeit	12
2.2.1.10 Transaktionszeit	13
2.2.1.11 Bitemporalität	14
2.2.1.12 Datenmodelle	14
2.2.2 Temporale Datenhaltung in herkömmlichen relationalen DBMS	14
2.2.2.1 Zeitstempelung	15
2.2.2.2 Primärschlüssel	16
2.2.2.3 Integritätsbedingungen	16
3 Anforderungsanalyse	19
3.1 Ist-Analyse	20
3.1.1 Das bisherige Verwaltungssystem	20
3.1.1.1 Archivierung	20
3.1.1.2 Aufbewahrungsfristen	22

3.1.1.3	Anpassungen	22
3.1.2	Die Verwaltungsanwendung	23
3.2	Anforderungen	24
3.2.1	(A) Allgemeine Anforderungen	25
3.2.1.1	(A1) Aufnahmegranularität	26
3.2.1.2	(A2) Direkter Zugriff	26
3.2.1.3	(A3) Verlustfreiheit	26
3.2.1.4	(A4) Benutzerveranlassung	27
3.2.1.5	(A5) Ordnungskriterien	27
3.2.1.6	(A6) Indexierung	27
3.2.1.7	(A7) Unabhängigkeit	27
3.2.1.8	(A8) Agilität	28
3.2.1.9	(A9) Metadaten	28
3.2.1.10	(A10) Zugriffsschutz	29
3.2.1.11	(A11) Unveränderbarkeit	29
3.2.1.12	(A12) Angemessenheit	29
3.2.2	(V) VGBS-Anforderungen	30
3.2.2.1	(V1) Jährlicher Schnappschuss	30
3.2.2.2	(V2) Verfassen	30
3.2.2.3	(V3) Bearbeiten	31
3.2.2.4	(V4) Löschen	31
3.3	Priorisierung	31
3.3.1	Definition des Entscheidungsproblems	31
3.3.2	Konstruktion der Hierarchie	31
3.3.3	Vergleich und Bewertung	32
3.3.4	Gewichtung und Lösung	34
3.4	Zusammenfassung	37
4	Vorstellung der Archivierungsmethoden	39
4.1	Begleitendes Beispiel	40
4.2	Datenbankinterne Methode	42
4.2.1	Konzepte	43
4.2.1.1	Verwaltungsjahr	43
4.2.1.2	VGBS-Gültigkeitszeit	43
4.2.1.3	Transaktionszeit	44
4.2.2	Datentrennung	45
4.2.3	Beispiel	45
4.3	Datenbankexterne Methode	49
4.3.1	Konzepte	50
4.3.1.1	Verwaltungsjahr	50
4.3.1.2	Modifikationszeit	50
4.3.2	Beispiel	50
4.4	Zusammenfassung	52
5	Evaluation	55
5.1	Vergleich	55
5.1.1	(A2) Direkter Zugriff	56
5.1.2	(A3) Verlustfreiheit	57

5.1.3	(A5) Ordnungskriterien	57
5.1.4	(A6) Indexierung	58
5.1.5	(A7) Unabhängigkeit	59
5.1.6	(A8) Agilität	59
5.1.7	(A9) Metadaten	60
5.1.8	(A10) Zugriffsschutz	61
5.1.9	(A11) Unveränderbarkeit	61
5.1.10	(A12) Angemessenheit	62
5.1.11	(V1-4) VGBS-Anforderungen	62
5.2	Entscheidung	63
6	Schlussbetrachtungen	65
6.1	Zusammenfassung	65
6.2	Ausblick	66
A	Anhang	69
A.1	Vergleich der Anforderungen	69
A.1.1	(A2) Direkter Zugriff	69
A.1.2	(A3) Verlustfreiheit	71
A.1.3	(A5) Ordnungskriterien	72
A.1.4	(A6) Indexierung	74
A.1.5	(A7) Unabhängigkeit	75
A.1.6	(A8) Agilität	77
A.1.7	(A9) Metadaten	78
A.1.8	(A10) Zugriffsschutz	79
A.1.9	(A11) Unveränderbarkeit	80
A.1.10	(A12) Angemessenheit	80
A.1.11	(V1) Jährlicher Schnappschuss	81
A.1.12	(V2) Verfassen	81
A.1.13	(V3) Bearbeiten	81
A.2	Funktionsweise der Archivierungsmethoden	81
A.2.1	Datenbankinterne Methode	82
A.2.2	Datenbankexterne Methode	87
	Literaturverzeichnis	93

Abbildungsverzeichnis

2.1	Generische Hierarchie [BR07]	7
2.2	Transaktionszeit am Beispiel	13
3.1	Das bisherige Verwaltungssystem	21
3.2	Archivierung im Kontext des bisherigen Verwaltungssystems.	22
3.3	Die neu-entwickelte Verwaltungsanwendung	24
3.4	Hierarchie	32
4.1	25.05.15 - 12:49:34 Uhr: Tim schreibt Danny in die Sportgruppe „SGKF“ ein.	41
4.2	08.04.16 - 12:45:56 Uhr: Tim vertuscht seinen Fehler im operativen Datenbestand.	42
4.3	Datenbankinterne Methode: Übersicht	42
4.4	Tim fügt Danny der Datenbank hinzu.	46
4.5	Ablauf der Geschehnisse	47
4.6	Datenbankexterne Methode: Übersicht	49
5.1	Hierarchie	56
A.1	Tim fügt Danny der Datenbank hinzu.	83
A.2	Ablauf der Geschehnisse	84
A.3	Löschen: Szenario 1	86
A.4	Löschen: Szenario 2	87
A.5	Wiederherstellen eines vorhergehenden Verwaltungsjahres	89
A.6	Wiederherstellen eines jüngeren Verwaltungsjahres	89
A.7	Datenbankschema des VGBS	91

Tabellenverzeichnis

2.1	Bewertungsskala der AHP-Methode [Saa08]	6
2.2	Vergleich von 3 Kriterien in Bezug auf das Entscheidungsziel.	8
2.3	Vergleich von 3 Subkriterien in Bezug auf Kriterium 1.	8
2.4	Vergleich von 3 Alternativen in Bezug auf das Subkriterium 1.	8
2.5	Spaltensummen von Tabelle 2.2	9
2.6	Spaltensummen von Tabelle 2.3	9
2.7	Spaltensummen von Tabelle 2.4	9
2.8	Lokale und globale Gewichtung der Kriterien in Bezug auf das Ziel.	9
2.9	Lokale und globale Gewichtung der Subkriterien in Bezug auf das Kriterium 1.	10
2.10	Lokale und globale Gewichtung der Alternativen in Bezug auf das Subkriterium 1.	10
2.11	Ergebnisse des Beispiels	10
2.12	MySQL Zeit-Datentypen	15
2.13	Datensätze mit drei alternativen Intervallenden	16
3.1	Übersicht der erteilten Bewertungen	33
3.2	Übersicht der Prioritäten in Prozent, sortiert nach ihrer Relevanz für den VGBS	34
5.1	Ergebnis des Vergleichs bzgl. „Direkter Zugriff“	57
5.2	Ergebnis des Vergleichs bzgl. „Verlustfreiheit“	57
5.3	Ergebnis des Vergleichs bzgl. „Ordnungskriterien“	58
5.4	Ergebnis des Vergleichs bzgl. „Indexierung“	58
5.5	Ergebnis des Vergleichs bzgl. „Unabhängigkeit“	59
5.6	Ergebnis des Vergleichs bzgl. „Agilität“	60

5.7	Ergebnis des Vergleichs bzgl. „Metadaten“	60
5.8	Ergebnis des Vergleichs bzgl. „Zugriffsschutz“	61
5.9	Ergebnis des Vergleichs bzgl. „Unveränderbarkeit“	62
5.10	Ergebnis des Vergleichs bzgl. „Angemessenheit“	62
5.11	Ergebnis des Vergleichs bzgl. der Anforderungen des VGBS	62
5.12	Übersicht der gewichteten Bewertungen sowie des Ergebnisses des Vergleichs	63

1. Einführung

Täglich fallen in Unternehmen, Vereinen oder anderen Institutionen beträchtliche Mengen von Daten an, die in einer Datenbank gespeichert werden müssen. Dabei geht der Trend hin zu immer größer werdenden digitalen Datenmengen. So soll laut einer Studie des „EMC“ das weltweite Datenvolumen bis 2020 um das Zehnfache anwachsen – von derzeit 4,4 Billionen Gigabyte auf 44 Billionen Gigabyte. Hierbei kann es z.B. aus rechtlichen oder aus Gründen des Datenerhalts von Produkten in Unternehmen, notwendig sein, diese Daten zu archivieren und für eine festgelegte Zeitdauer zugänglich zu halten. Generell betrachtet bedeutet Archivierung, die kontrollierte und systematische langfristige Speicherung von Dokumenten und Daten [VER14]. Im Kontext digitaler Daten hat sich in diesem Zusammenhang besonders der Begriff der elektronischen Archivierung durchgesetzt. Hierfür existieren in Deutschland landläufig 2 Definitionen:

Zum einen meint elektronische Archivierung „die datenbankgestützte, langzeitige, sichere und unveränderbare Aufbewahrung von jederzeit wieder reproduzierbaren elektronischen Informationsobjekten“ [Kam06a].

Zum anderen wird hierbei aber auch von einer revisionssicheren Archivierung gesprochen. Revisionssichere Archivierung bedeutet, dass Archivierungssysteme bestimmten rechtlichen Vorgaben entsprechen müssen. Darunter befinden sich die Vorgaben des Handelsgesetzbuches (HGB)(§239, §257), der Abgabenordnung (AO)(§146, §147) sowie den Grundsätzen zur ordnungsmäßigen Führung und Aufbewahrung von Büchern, Aufzeichnungen und Unterlagen in elektronischer Form sowie zum Datenzugriff (GoBD) [Kam06b].

Auch das Archivierungssystem des Vereins für Gesundheit, Bewegung und Sport (VGBS) muss diesen Vorgaben gerecht werden. Der VGBS ist ein gemeinnütziger Verein mit Sitz in Magdeburg, welcher sich dem Rehabilitatssport verschrieben hat. Als Verein ist der VGBS dazu verpflichtet sämtliche jahresbezogenen Daten seiner Mitglieder, Nichtmitglieder sowie Kursteilnehmer aufzubewahren und zugänglich zu halten. Zum einen benötigt der Verein diese Daten, damit er Einnahmen aus seinen Sportgruppen und Kursen im Falle einer steuerrechtlichen Außenprüfung ordnungs-

gemäß nachweisen kann und zum anderen um gerichtlichen Auseinandersetzungen nachkommen zu können.

Die aktuelle Archivierungsmethode des VGBS besteht darin, die 3 Microsoft Excel-Dateien, welche den Kern ihres Datenverwaltungssystems bilden, am Jahresende bzw. -beginn zu kopieren und für das kommende Jahr neu anzulegen. Dabei werden sämtliche jahresbezogenen Daten wie beispielsweise gezahlte monatliche Beiträge oder geführte Anwesenheitslisten, aus den neu angelegten Dateien gelöscht. Im Anschluss werden die alten Dateien archiviert und die kopierten für den operativen Betrieb genutzt.

Auf Grund von Redundanzen, Inkonsistenzen, fehlender Mehrbenutzerfähigkeit und anderer Probleme bei der Verwendung dieser Dateien für die Verwaltung seiner operativen Daten entschied sich der Verein sein Datenverwaltungssystem komplett neu zu entwickeln. Dieses neu entwickelte Verwaltungssystem besteht aus zwei Grundpfeilern: Einer auf einem Webserver liegenden, MySQL-Datenbank und einer webgestützten Datenbankanwendung, welche darauf aufbaut.

Einerseits werden durch diese Anwendung zwar ein Großteil der Verwaltungsprobleme des VGBS gelöst, andererseits entstanden neue Herausforderungen im Bezug auf die Archivierung der Vereinsdaten. So existieren zwar eine Vielzahl verschiedener Methoden und Konzepte, um Daten von relationalen Datenbanken zu archivieren, jedoch wird diese Auswahl durch die Anforderungen die an das Archiv des VGBS gestellt werden beschränkt:

Dabei stehen auf der einen Seite allgemeine Anforderungen, welche von verschiedenen Autoren an ein digitales Archiv gestellt werden. Unter diesen befindet sich auch die rechtliche Forderung nach einer Unveränderbarkeit der archivierten Daten nach § 146 Abs. 4 AO und § 239 HGB. Unveränderbarkeit bedeutet in diesem Kontext, dass der Ursprungszustand und alle Modifikationen der Daten über die Dauer der Aufbewahrungsfrist nachvollziehbar sind [dL14].

Auf der anderen Seite steht jedoch die Forderung des Vereins seine archivierten Daten verändern zu können, um beispielsweise nachträglich gezahlte Beiträge auch in das Archiv aufzunehmen. Dies widerspricht jedoch der Unveränderbarkeit.

Ferner existieren zur Archivierung der Vereinsdaten grundlegend zwei Ansätze: Einerseits den datenbankinternen Ansatz, bei dem die archivierten Daten innerhalb des DBS aufbewahrt werden. Andererseits den datenbankexternen Ansatz, im Kontext dessen diese in Dateien außerhalb des DBS gespeichert werden. Um mit Hilfe dieser Ansätze den oben beschriebenen Widerspruch aufzulösen, wird es erforderlich eigene Archivierungsmethoden zu entwickeln, welche auf diese besonderen Anforderungen im Kontext des Vereins zugeschnitten sind.

Zielstellung der Arbeit

Die primäre Zielstellung der Arbeit ist es ein Konzept zur Archivierung für den Verein zu entwickeln. Da es zwei grundsätzliche Ansätze gibt, soll mit Hilfe einer Priorisierung der Anforderungen und Bewertung festgestellt werden, welche besser geeignet ist. Dabei ist es auch erforderlich den bereits angedeuteten Widerspruch konzeptuell zu lösen.

Gliederung der Arbeit

Die Arbeit ist wie folgt aufgebaut. In Kapitel 2 werden die Grundlagen, die für das Verständnis dieser Arbeit essentiell sind erläutert. In diesem Zusammenhang stelle zunächst ich den Analytic Hierarchy Process (AHP) vor. Dieser wird mir im späterem Verlauf dieser Arbeit dienen die in Kapitel 3 identifizierten Anforderungen zu priorisieren sowie die Ergebnisse des Vergleichs der datenbankexternen und -internen Methode zu quantifizieren. Weiterhin werde ich nachfolgend die temporale Datenhaltung, sowie hierfür wichtige Konzepte vorstellen. Diese Konzepte dienen mir bei der Umsetzung der datenbankinternen Methode.

Das Kapitel 3 beschäftigt sich mit der Analyse der technischen Anforderungen, welche an die Archivierung der Vereinsdaten gestellt werden sowie deren Priorisierung. Im Rahmen dessen werde ich als erstes eine Ist-Analyse der Verwaltung des VGBS durchführen. Darauf aufbauend werde ich die Anforderungen zum einen aus Fachliteratur und gesetzlichen Vorgaben zum Thema der digitalen Archivierung ableiten. Andererseits werden jene Anforderungen identifiziert, die direkt vom Verein gestellt werden. Im Rahmen der Priorisierung findet dann der AHP Anwendung. Hierfür werden die Anforderungen paarweise miteinander verglichen und bewertet.

Im Nachfolgendem widmet sich Kapitel 4 der Vorstellung der datenbankinternen und datenbankexternen Methode. Im Zuge dessen wird hierfür zunächst ein Beispiel eingeführt. Dieses dient als Referenz für die Fähigkeiten der beiden Methoden und wird deren Vorstellung begleiten. Darauffolgend werden die datenbankinterne sowie datenbankexternen Methode vorgestellt. Im Rahmen dessen wird deren Funktionsweise aufgezeigt und beschrieben, wie der bereits angedeutete Widerspruch aus Unveränderbarkeit sowie Veränderbarkeit von beiden Methoden konzeptuell gelöst wird.

In Kapitel 5 findet der Vergleich der datenbankinternen mit der datenbankexternen Methode statt. Hierfür wird ein qualitativer Vergleich, anhand der im Kapitel 3 analysierten Anforderungen, vorgenommen. Im Rahmen dessen wird jeweils verglichen sowie bewertet wie gut eine Methode eine Anforderung erfüllt. Für die Bewertung der beiden Methoden im Zuge des Vergleichs findet wieder die in Kapitel 2 vorgestellte AHP-Methode Anwendung. Hiermit wird die Quantifizierung der Vergleichsergebnisse vorgenommen sowie ein Index errechnet anhand dessen konkret erkennbar ist, welche Methode im Hinblick auf die analysierte Anforderungen für den VGBS am besten geeignet ist. Unter zu Hilfenahme der Vergleichsergebnisse und dieses Index werde ich ein Fazit ziehen und mich letztlich für eine Methode entscheiden.

In Kapitel 6 fasse ich die Ergebnisse dieser Arbeit noch einmal zusammen und gebe einen Ausblick auf zukünftige Entwicklungen.

2. Grundlagen

In diesem Kapitel werde ich die Grundlagen erläutern, welche für das Verständnis dieser Arbeit essentiell sind. Dafür werde ich als erstes den Analytic Hierarchy Process (AHP) vorstellen. Dieser ist eine Methode um komplexe Entscheidungsprozesse zu unterstützen. Der AHP wird mir bei der im folgenden Kapitel 3 stattfindenden Priorisierung der Anforderungen, sowie in Kapitel 5 beim Vergleich der datenbankexternen und datenbankinternen Methode behilflich sein. Im Anschluss daran stelle ich die temporale Datenhaltung vor. Dessen Konzepte bilden die Grundlage für die in Kapitel 4 vorgestellte datenbankinterne Methode.

2.1 Analytic Hierarchy Process

Der nachfolgende Abschnitt beschäftigt sich mit dem AHP. Die AHP-Methode wurde von Thomas L. Saaty entwickelt und hilft dabei komplexe Entscheidungen zu treffen. Der Analytic Hierarchy Process wird im Verlauf dieser Arbeit genutzt um die Anforderungen an das digitale Archiv des VGBS zu priorisieren sowie die Archivierungsmethoden hinsichtlich dieser zu bewerten. Die AHP-Methode zeichnet sich durch besonders hohe Zuverlässigkeit und Fehlertoleranz in Bezug auf die Priorisierung von Anforderungen aus, weil diese nicht für sich alleine, sondern immer im Kontext der anderen Anforderungen bewertet werden [\[KWR98\]](#).

Der Name Analytic Hierarchy Process leitet sich von der Struktur dieser Methode ab. Hierbei wird ein komplexes Entscheidungsproblem in eine Hierarchie bestehend aus der Problemstellung, den Kriterien und Subkriterien, die dabei helfen dieses Problem zu lösen, sowie in die zu vergleichenden Alternativen unterteilt. Ferner wird der Kern der AHP-Methode dadurch gebildet, dass zwei der Elemente dieser Hierarchie paarweise miteinander verglichen und mit Hilfe einer 9-Stufigen Skala bewertet werden. Die Bewertung erfolgt dabei rein subjektiv nach der Wichtigkeit der verglichenen Elemente in Bezug auf das in der Hierarchie nächst höher liegende Element [\[Saa08, BR07\]](#).

2.1.1 Bewertungsskala

Für den Vergleich von Elementen im Rahmen des Analytic Hierarchy Process ist zunächst eine Skala entscheidend. Diese gibt Aufschluss darüber, wie viel wichtiger ein Element verglichen zu einem anderen, in Bezug zu einem Vergleichskriterium, ist [Saa08]. Diese Skala wird von Saaty in [Saa08] definiert und ist in Tabelle 2.1 illustriert. Diese ist aufsteigend nach dem Grad der Wichtigkeit sortiert und beginnt mit 1. Bekommt ein Element den Wert 1 zugeteilt ist dieses von gleicher Wichtigkeit, wie das Element mit dem es verglichen wurde. Nachfolgend steigt der Grad der Wichtigkeit eines Elements, verglichen zu einem anderen, von 3: das Element ist moderat wichtiger, über 5: es ist von besonders großer Wichtigkeit, bis hin zu 9: das Element ist extrem viel wichtiger. Die dazwischenliegenden Werte beschreiben stets eine höhere Wichtigkeit, als der vorhergehende Wert. Ist hingegen ein Element weniger wichtig, als ein anderes bekommt es den äquivalenten reziproken Wert dieser Skala zugewiesen. Weiterhin lassen sich noch feinere Unterschiede in der Wichtigkeit durch Kommastellen ausdrücken.

Grad der Wichtigkeit	Definition
1	Gleich wichtig
2	Schwach oder etwas wichtig
3	Moderate Wichtigkeit
4	Moderat+
5	Große Wichtigkeit
6	Groß+
7	Sehr große Wichtigkeit
8	Sehr sehr groß
9	Extreme Wichtigkeit
Reziproke dieser Werte	Wenn ein Element i, verglichen zu einem Element j, eine der oben beschriebenen Werte zugeteilt wird, dann erhält j den reziproken Wert, wenn es mit i verglichen wird
1,0-1,9	Wenn Elemente, in Bezug auf ihre Wichtigkeit, sehr nah bei einander liegen

Tabelle 2.1: Bewertungsskala der AHP-Methode [Saa08]

2.1.2 Ablauf

In Anlehnung an [MK13, Saa08] untergliedert sich der Analytic Hierarchy Process, wie nachfolgend erläutert, in 4 Schritte:

2.1.2.1 Definition des Entscheidungsproblems

Zunächst erfolgt die Definition des Entscheidungsproblems. Im Rahmen dessen werden alle Daten gesammelt, die für die Lösung dieses Problems relevant sind. Aus diesen Daten werden Kriterien abgeleitet, mit deren Hilfe dieses Problem gelöst werden soll. Je nach Komplexität können sich diese weiter in Subkriterien untergliedern [MK13]. Im Anschluss daran werden die Alternativen festgelegt, welche der Lösung des Entscheidungsproblems dienen.

2.1.2.2 Konstruktion der Hierarchie

In diesem Schritt erfolgt die Erstellung der Hierarchie bestehend aus dem Ziel der Entscheidung und den, im vorhergehenden Schritt festgelegten, Kriterien, Subkriterien sowie den Alternativen. Dabei bildet das Ziel der Entscheidung die oberste Hierarchieebene. Diese wird gefolgt von den Kriterien, den Subkriterien und auf der untersten Ebene den Alternativen [MK13]. Dabei beeinflusst jede Ebene die darunterliegende. In [Abbildung 2.1](#) wird eine generische Hierarchie eines Entscheidungsproblems dargestellt.

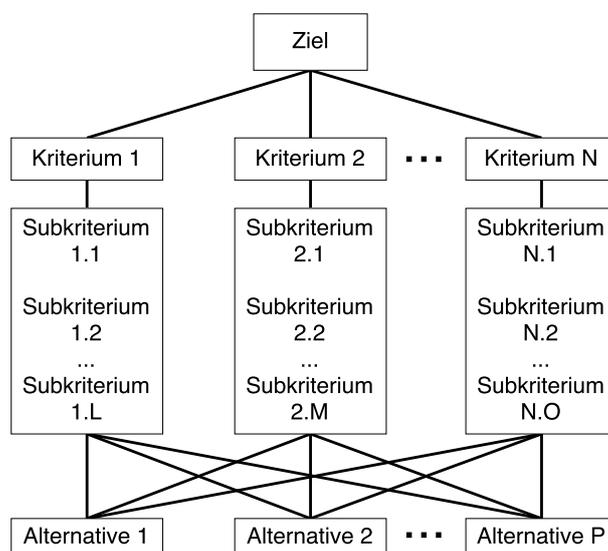


Abbildung 2.1: Generische Hierarchie [BR07]

2.1.2.3 Vergleich und Bewertung

Im Kontext dieses Schritts werden die Elemente paarweise verglichen und bewertet [Saa08]. Diesen sowie die folgenden Schritte des AHP werde ich exemplarisch an einem Zweig der Hierarchie aus [Abbildung 2.1](#) illustrieren. Die Hierarchie wird für dieses Beispiel aus 3 Kriterien, mit jeweils 3 Subkriterien und 3 Alternativen bestehen.

Ferner werden nur die Elemente einer Ebene miteinander verglichen. Dabei stellen die Elemente der nächst höheren Ebene die Kriterien dar anhand derer dieser Vergleich stattfindet [MK13, Saa08]. Für diesen Schritt werden in der Regel für jedes Vergleichskriterium Matrizen erstellt, welche die zu vergleichenden Elemente als Zeilen- und Spaltenwerte besitzen [MK13]. Siehe Tabelle 2.2, hier werden die übergeordneten Kriterien in Bezug auf das Ziel verglichen. In Tabelle 2.3 findet der Vergleich der Subkriterien 1.1-1.3, in Bezug auf ihr übergeordnetes Element dem Kriterium 1, statt. Tabelle 2.4 beschreibt den Vergleich der Alternativen relativ zum Subkriterium 1. Die weiteren Vergleiche der Subkriterien von Kriterium 2 und 3, sowie die verbleibenden Vergleiche der Alternativen in Bezug auf diese finden natürlich im Kontext des AHP statt, werden jedoch für dieses Beispiel nicht dargestellt.

Im Rahmen des Vergleichs wird nun sukzessiv jedes Element der ersten Spalte mit jedem Element in der ersten Zeile verglichen und bewertet. Der Entscheider bewertet hierbei, welche der verglichenen Elemente ihm wie viel wichtiger in Bezug auf das Vergleichskriterium erscheint.

Dafür vergibt er Werte aus der in Tabelle 2.1 dargestellten Skala. In Tabelle 2.2 ist Kriterium 1 etwas wichtiger, als Kriterium 2 und erhält somit den Wert 2. Ferner umfasst die Diagonale einer Vergleichsmatrix dabei stets nur den Wert 1, weil hierbei die selben Elemente miteinander verglichen werden und diese als gleich wichtig betrachtet werden. Weiterhin geht der AHP beim Vergleich eines Elements i mit dem Element j davon aus, dass der Entscheider beim umgekehrten Vergleich von j und i zur entgegengesetzten Bewertung kommen würde [MK13]. Deshalb kann die untere Hälfte der Vergleichsmatrix unter der Diagonale mit den reziproken der Werte über der Diagonale gefüllt werden.

	Kriterium 1	Kriterium 2	Kriterium 3
Kriterium 1	1	2	5
Kriterium 2	1/2	1	1/6
Kriterium 3	1/5	1/6	1

Tabelle 2.2: Vergleich von 3 Kriterien in Bezug auf das Entscheidungsziel.

	Subkriterium 1.1	Subkriterium 1.2	Subkriterium 1.3
Subkriterium 1.1	1	1/4	9
Subkriterium 1.2	4	1	3
Subkriterium 1.3	1/9	1/3	1

Tabelle 2.3: Vergleich von 3 Subkriterien in Bezug auf Kriterium 1.

	Alternative 1	Alternative 2	Alternative 3
Alternative 1	1	5	1
Alternative 2	1/5	1	1/2
Alternative 3	1	2	1

Tabelle 2.4: Vergleich von 3 Alternativen in Bezug auf das Subkriterium 1.

2.1.2.4 Gewichtung und Lösung

Im Rahmen dieses vierten und letzten Schritts werden die Bewertungen aus dem vorhergehenden Schritt genutzt, um die Gewichtung der Elemente der Hierarchie vorzunehmen. Das Ergebnis dieses Schritts ist ein Index der beschreibt, welche Alternative die beste für die Lösung des Entscheidungsproblems ist.

Hierfür wird als erstes die lokale Gewichtung der Elemente berechnet. Diese illustriert wie Wichtig ein Element im Bezug auf das nächst höher gelegene Element ist. Dazu werden zunächst die Spaltensummen der jeweiligen Vergleichsmatrix gebildet (siehe [Tabelle 2.5](#), [Tabelle 2.6](#), [Tabelle 2.7](#)). Im nachfolgenden werden die Bewertungen aus dem paarweisen Vergleich durch ihre entsprechende Spaltensumme dividiert. Dadurch werden die Vergleichsmatrizen auf 1 normiert.

	Kriterium 1	Kriterium 2	Kriterium 3
Kriterium 1	1	2	5
Kriterium 2	1/2	1	1/6
Kriterium 3	1/5	6	1
Spaltensumme	1,7	9	6,17

Tabelle 2.5: Spaltensummen von [Tabelle 2.2](#)

	Subkriterium 1.1	Subkriterium 1.2	Subkriterium 1.3
Subkriterium 1.1	1	1/4	9
Subkriterium 1.2	4	1	3
Subkriterium 1.3	1/9	1/3	1
Spaltensumme	5,11	1,58	13

Tabelle 2.6: Spaltensummen von [Tabelle 2.3](#)

	Alternative 1	Alternative 2	Alternative 3
Alternative 1	1	5	1
Alternative 2	1/5	1	1/2
Alternative 3	1	2	1
Spaltensumme	2,2	8	2,5

Tabelle 2.7: Spaltensummen von [Tabelle 2.4](#)

	Kriterium 1	Kriterium 2	Kriterium 3	Lokale Gew.	Globale Gew.
Kriterium 1	$1/1,7 = 0,59$	$2/9 = 0,22$	$5/6,17 = 0,81$	$(0,59+0,22+0,81)/3 = 0,54$	0,54
Kriterium 2	$4/1,7 = 0,29$	$1/9 = 0,11$	$(1/6)/6,17 = 0,03$	$(0,29+0,11+0,03)/3 = 0,14$	0,14
Kriterium 3	$(1/9)/1,7 = 0,12$	$6/9 = 0,67$	$1/6,17 = 0,16$	$(0,12+0,67+0,16)/3 = 0,32$	0,32

Tabelle 2.8: Lokale und globale Gewichtung der Kriterien in Bezug auf das Ziel.

	Subkr. 1.1	Subkr. 1.2	Subkr. 1.3	Lokale Gew.	Globale Gew.
Subkr.1.1	$1/5,11 = 0,2$	$(1/4)/1,58 = 0,16$	$9/13 = 0,69$	$(0,2+0,16+0,69)/3 = 0,35$	$0,35*0,54 = 0,19$
Subkr.1.2	$4/5,11 = 0,78$	$1/1,58 = 0,63$	$3/13 = 0,23$	$(0,78+0,63+0,23)/3 = 0,55$	$0,55*0,54 = 0,3$
Subkr. 1.3	$1/5,11 = 0,02$	$(1/3)/1,58 = 0,21$	$1/13 = 0,08$	$(0,02+0,21+0,08)/3 = 0,1$	$0,1*0,54 = 0,05$

Tabelle 2.9: Lokale und globale Gewichtung der Subkriterien in Bezug auf das Kriterium 1.

	Alternative 1	Alternative 2	Alternative 3	Lokale Gew.	Globale Gew.
Alternative 1	$1/2,2 = 0,46$	$5/8 = 0,63$	$1/2,5 = 0,4$	$(0,46+0,63+0,4)/3 = 0,5$	$0,5*0,19 = 0,1$
Alternative 2	$(1/5)/2,2 = 0,09$	$1/8 = 0,13$	$(1/2)/2,5 = 0,2$	$(0,09+0,13+0,2)/3 = 0,14$	$0,14*0,19 = 0,03$
Alternative 3	$1/2,2 = 0,46$	$2/8 = 0,25$	$1/2,5 = 0,4$	$(0,46+0,25+0,4)/3 = 0,37$	$0,37*0,19 = 0,07$

Tabelle 2.10: Lokale und globale Gewichtung der Alternativen in Bezug auf das Subkriterium 1.

Im nächsten Schritt werden die Zeilenelemente der normierten Vergleichsmatrizen zu Zeilensummen aufaddiert und diese durch die Anzahl der Zeilenelemente dividiert. Die daraus resultierenden Werte bilden nun die lokale Gewichtung der Hierarchieelemente, welche ebenfalls den Eigenvektor der jeweiligen Vergleichsmatrix darstellt [MK13]. Anschließend daran wird die globale Gewichtung berechnet, indem die Gewichtung der Elemente einer Hierarchieebene mit der Gewichtung des Elements in der nächst höheren Ebene multipliziert werden. Die Globale Gewichtung beschreibt wie Wichtig ein Element zur Lösung des Entscheidungsproblems ist. Die Ergebnisse dieses Prozesses werden in Tabelle 2.8, Tabelle 2.9 und Tabelle 2.10 dargestellt.

Um jetzt die beste Alternative auszuwählen werden sämtliche Gewichtungen der Alternativen, in Bezug auf die Kriterien und Subkriterien, aufaddiert. Dadurch entsteht ein Index der darstellt mit welchem Gewicht die einzelnen Alternativen zur Lösung des Entscheidungsproblems beitragen, wodurch die beste Alternative ausgewählt werden kann [MK13]. In Tabelle 2.11 sind die Ergebnisse des AHP im Rahmen des Beispiels dargestellt. Die erste Spalte beinhaltet die Ergebnisse aus Tabelle 2.10. Im Gegensatz dazu ergeben sich die zusätzlichen Werte aus den Vergleichen die in diesem Beispiel nicht behandelt worden. Addiert man nun pro Zeile jeden Spaltenwert auf erhält man den Wert in der Spalte „Zeilensumme“. Aus dieser ist klar erkennbar, dass Alternative 1 mit 0,385 die beste ist um das Entscheidungsproblem zu lösen.

Alt.	Subkrit. 1.1	Subkrit. 1.2	Subkrit. 1.3	Subkrit. 2.1	Subkrit. 2.2	Subkrit. 2.3	Subkrit. 3.1	Subkrit. 3.2	Subkrit. 3.3	Zeilensum.
Alt. 1	0,1	0,12	0,02	0,01	0,03	0,005	0,05	0,02	0,03	0,385
Alt. 2	0,03	0,12	0,01	0,02	0,02	0,005	0,04	0,03	0,03	0,305
Alt. 3	0,07	0,06	0,02	0,02	0,03	0,01	0,01	0,06	0,03	0,31
Spaltensum.	0,2	0,3	0,05	0,05	0,08	0,02	0,1	0,11	0,09	1

Tabelle 2.11: Ergebnisse des Beispiels

2.2 Temporale Datenhaltung

Die temporale Datenhaltung ist besonders relevant im Kontext der datenbankexternen Methode. Diese verwendet die in der temporalen Datenhaltung inbegriffenen Konzepte der Gültigkeitszeit und Transaktionszeit. Unter Temporaler Datenhaltung versteht man das Festhalten der zeitlichen Entwicklung von Daten in einer Datenbank. Hierbei können Veränderungen der Daten über die Zeit gespeichert und verfolgt werden. Dazu werden die zu verfolgenden Daten um einen zeitlichen Bezug ergänzt. Bei der temporalen Datenhaltung sind im wesentlichen 3 zeitliche Konzepte von zentraler Bedeutung: Die Gültigkeitszeit und Transaktionszeit sowie die Bitemporalität [JDB⁺98]. In diesem Zusammenhang lassen sich mit jedem dieser Konzepte verschiedene zeitliche Sachverhalte abbilden. Die Gültigkeitszeit zeigt auf wann ein Fakt in der realen Welt als wahr betrachtet wird. Dahingegen beschreibt die Transaktionszeit wann ein Fakt in der Datenbank aktuell war. Werden beide Konzepte benötigt spricht man von der Bitemporalität.

Der nachfolgende Abschnitt ist wie folgt gegliedert: Zunächst werde ich zentrale Konzepte und Begriffe der temporalen Datenhaltung erläutern. Im Zuge dessen, werde ich drei Datenmodelle, welche zur Umsetzung der oben genannten Konzepte entwickelt wurden, vorstellen. Im Anschluss daran werde ich erläutern wie die temporale Datenhaltung in herkömmlichen relationalen DBMS umgesetzt werden kann.

2.2.1 Zentrale Konzepte

In diesem Abschnitt werde ich grundlegende Konzepte sowie Begriffe der temporalen Datenhaltung vorstellen. Diese orientieren sich im wesentlichen an denen in [JDB⁺98] definierten Konzepten und Begriffen.

2.2.1.1 Granularität

Unter Granularität versteht man die Feinheit einer Zeitangabe.

2.2.1.2 Chronon

Als Chronon bezeichnet man das kleinste mögliche Zeitintervall bei einer bestimmten Granularität. Bei einem Datum wäre ein Chronon beispielsweise ein Tag.

2.2.1.3 Zeitpunkt

Ein Zeitpunkt beschreibt einen Punkt auf einer gewählten Zeitachse. Dieser kann in unterschiedlicher Granularität vorliegen.

2.2.1.4 Zeitintervall

Zeitintervalle werden durch zwei Zeitpunkte beschrieben. Sie können verankert oder unverankert sein. Verankerte Zeitintervalle beschreiben konkrete Intervalle auf einer Zeitskala, wohingegen unverankerte Zeitintervalle eine Zeitdauer darstellen. Im mathematischen Sinn unterscheidet man hierbei zwischen beschränkten und unbeschränkten Intervallen.

beschränktes Intervall

Ein beschränktes Intervall wird als abgeschlossen bezeichnet, wenn es beide Grenzen enthält und als offen wenn keine der Grenzen im Intervall enthalten sind. Zusätzlich lassen sich rechtsseitig halboffene und linksseitig halboffene beschränkte Intervalle unterscheiden. Um ein rechtsseitig halboffenes Intervall handelt es sich, wenn die rechte Grenze nicht im Intervall enthalten ist, wobei bei einem linksseitigen halboffenen Intervall die linke Grenze nicht im Intervall enthalten ist.

unbeschränktes Intervall

Im Vergleich zu einem beschränkten Intervall, spricht man von einem unbeschränkten Intervall, wenn mindestens eine oder beide Intervallgrenzen fehlen.

2.2.1.5 Temporales Element

Ein temporales Element beschreibt eine endliche Menge von n Zeitintervallen.

2.2.1.6 Zeitstempelung

Unter Zeitstempelung versteht man das Hinzufügen eines zeitlichen Kontextes zu einem Datensatz. Dabei kann es sich um einen konkreten Zeitpunkt oder wie in den meisten Fällen um ein Zeitintervall handeln.

2.2.1.7 Schnappschuss

Als Schnappschuss oder im englischen Snapshot, bezeichnet man das Betrachten der temporalen Daten zu einem fixierten Zeitpunkt.

2.2.1.8 Benutzerdefinierte Zeit

Die benutzerdefinierte Zeit ist ein zeitliches Konzept, das im Kontext der temporalen Datenhaltung nicht interpretiert wird [JDB⁺98]. Ein Beispiel hierfür ist das Geburtsdatum einer Person.

2.2.1.9 Gültigkeitszeit

Sie beschreibt die Zeit zu der ein Fakt in der realen Welt als wahr bzw. gültig betrachtet wird [JDB⁺98]. Vor dem Hintergrund, dass Informationssysteme, wie beispielsweise die neu entwickelte Verwaltungsanwendung des VGBS, keine integrierten Möglichkeiten besitzen, Gegebenheiten aus der Realwelt eigenständig zu erfassen, müssen Gültigkeitszeiten vom Benutzer selbst bereitgestellt werden [Myr05]. Somit können sie vergangene, gegenwärtige oder zukünftige Zeiträume umfassen. Ein Datensatz, dessen Gültigkeitsende nicht festgelegt wurde, ist auf unbestimmte Zeit gültig. Im Zusammenhang mit diesem Konzept wäre ein Beispiel ein in der Datenbank eingetragener Datensatz der beschreibt, über welchen Zeitraum der zweite Weltkrieg stattfand. Das zeitliche Intervall von 1939 bis 1945 stellt dabei die Gültigkeitszeit dieses Datensatzes dar.

2.2.1.10 Transaktionszeit

Die Transaktionszeit bildet die Zeit ab, zu der ein Datensatz in der Datenbank als aktuell angesehen wird [JDB⁺98]. Aktuell bedeutet, dass ein in der Datenbank befindlicher Fakt als korrektes Abbild der Realität aufgefasst wird [Myr05]. Sie beschreibt, wie die Datenbank zu einem bestimmten Zeitpunkt beschaffen war und bildet die Korrekturhistorie der Daten ab. Die Transaktionszeit wird ausschließlich vom System vergeben und beschränkt sich dabei stets auf die Gegenwart [Myr05]. Das Einfügen eines nicht mehr oder noch aktuell werdenden Datensatzes ist dadurch ausgeschlossen.

Ferner beginnt die Transaktionszeit immer mit dem Einfügen eines Datensatzes und endet nur mit dessen Modifikation oder logischer Löschung [Myr05]. Im Falle einer Modifikation wird die veränderte Version dieses Datensatzes der Tabelle hinzugefügt und ihr ein aktueller Transaktionszeitstempel gegeben. Die ursprüngliche Version gilt als nicht mehr aktuell; befindet sich jedoch noch in der Datenbank. Ein nachträgliches Verändern, eines nicht mehr aktuellen Datensatzes, ist nicht möglich [JDB⁺98]. Weiterhin wird ein Datensatz, dessen Ende seiner Transaktionszeit noch nicht festgelegt wurde, als aktuell bezeichnet.

Ich möchte dies in [Abbildung 2.2](#) an einem Beispiel illustrieren:

Der VGBS speichert Denny am **31.02.16** in seine Datenbank ein. Dementsprechend wird der Beginn seiner Transaktionszeit auf dieses Datum festgelegt. Dahingegen ist das Ende noch offen, weil Denny noch nicht bearbeitet oder gelöscht wurde. Aus diesem Grund ist Denny noch aktuell bzw. gilt als korrektes Abbild der Realität. Am **26.07.16** stellt der Verein jedoch fest, dass ein Rechtschreibfehler vorliegt und korrigiert Denny zu Danny. In Folge dessen wird das Ende von Dennys Transaktionszeit auf den Zeitpunkt seiner Korrektur gesetzt und Dennys korrigierte Version Danny der Tabelle hinzugefügt. Der Beginn von Dannys Transaktionszeit wird hierbei ebenfalls auf den **26.07.16** festgelegt. Zusammen mit Dannys unbestimmten Ende seiner Transaktionszeit bedeutet dies, dass Danny ab diesem Zeitpunkt aktuell ist. Dahingegen ist Denny, durch das festlegen des Endes seiner Transaktionszeit, dies nicht mehr. Am **19.12.16** löscht der Verein Danny. Damit wird das Ende seiner Transaktionszeit auf diesen Zeitpunkt festgelegt.

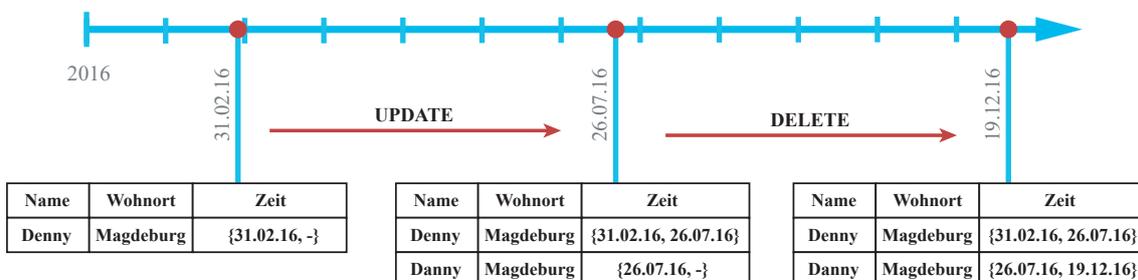


Abbildung 2.2: Transaktionszeit am Beispiel

2.2.1.11 Bitemporalität

Die Bitemporalität vereint die Konzepte der Gültigkeitszeit und Transaktionszeit. Damit kann nicht nur ausgedrückt werden, wann ein Fakt in der Realität als wahr angenommen, sondern auch wann dieser in der Datenbank erfasst und gelöscht wurde.

2.2.1.12 Datenmodelle

Zur Unterstützung der Gültigkeitszeit, Transaktionszeit und insbesondere der Bitemporalität wurden in der Vergangenheit einige Entwürfe vorgestellt, welche im wesentlichen auf der Erweiterung des relationalen Modells beruhen [Myr05]. Hierbei besitzen diese häufig eine eigene Datenstruktur sowie Datenbanksprache [Myr05]. In diesem Kontext wäre zunächst das Interval Extended Relational Model (IXRM) genannt [LM97]. Die hierzu gehörige Datenbanksprache IXSQL stellt eine Erweiterung von SQL dar. Hierfür wurde SQL um die Möglichkeit erweitert unterschiedliche Intervall-Daten zu behandeln. Hierzu zählen auch zeitliche Intervalle, welche zur Abbildung der oben genannten Konzepte verwendet werden können.

Ein weiterer Ansatz stellt die Sprache TSQL2 dar [SAA⁺94]. TSQL2 wurde von einer Vielzahl von Forschern entwickelt und basiert auf dem Bitemporal Conceptual Data Model. Die Zeitstempelung erfolgt hierbei implizit über bitemporale Elemente. Implizit bedeutet, dass im Zusammenhang mit dem BCDM keine aus relationalen Datenbanken bekannten Attribute zur Zeitstempelung herangezogen werden.

Über dies ist in diesem Kontext noch ATSQL2 zu erwähnen. ATSQL2 wurde von Steinmeier in [Ste98] vorgestellt und stellt im Gegensatz zu den bereits erwähnten Datenmodellen IXSQL und TSQL2, nur ein Frontend eines gewöhnlichen relationalen DBMS dar. Auf diesem können Datenbank-Anweisungen, welche an der Syntax von TSQL2 angelehnt sind, ausgeführt werden. In diesem Kontext ist auch TimeDB zu erwähnen, welche einen funktionsfähigen Prototypen von ATSQL2 darstellt.

Hiermit seien nur eine kleine Auswahl von Konzepten zur Umsetzung der temporalen Datenhaltung genannt. Außer für ATSQL2 existieren jedoch für keine dieser Konzepte funktionsfähige produktiv einsetzbare Systeme. Aus diesem Grund wird es zur Umsetzung der Gültigkeitszeit, Transaktionszeit und damit auch der Bitemporalität erforderlich diese in herkömmlichen relationalen DBMS abzubilden.

2.2.2 Temporale Datenhaltung in herkömmlichen relationalen DBMS

In herkömmlichen relationalen Datenbanken werden Gültigkeitszeit und Transaktionszeit nur in sofern unterstützt, dass diese unter Verwendung „normaler“ Datenbank-Attribute abgebildet werden. Hierbei werden die zu historisierenden Tabellen um entsprechende zeitliche Attribute ergänzt, mit deren Hilfe die Zeitstempelung der Tupel stattfindet. Jedes relationale DBMS bietet hierfür die Möglichkeit Attribute mit verschiedenen Zeit-Datentypen zu versehen. Beispielsweise bietet MySQL hierfür eine Reihe möglicher Datentypen in unterschiedlicher Granularität, welche in Tabelle 2.12 dargestellt sind.

Typ	Format	Beispiel
DATE	YYYY-MM-DD	2016-06-10
TIME	hh:mm:ss	12:55:45
DATETIME/TIMESTAMP	YYYY-MM-DD hh:mm:ss	2016-06-10 12:58:25
YEAR	YYYY	2016

Tabelle 2.12: MySQL Zeit-Datentypen

Ob die Zeitstempelung eines Tupels auf Basis eines Zeitintervalls oder Zeitpunkts geschieht ist dem Datenbankdesigner überlassen.

2.2.2.1 Zeitstempelung

Zeitpunkt

Entfällt die Entscheidung des Datenbankdesigners, welche Form der Zeitstempelung er verwendet auf einen Zeitpunkt, so werden die dazugehörigen Tabellen um ein Zeitattribut ergänzt. Dieses stellt den Beginn des Zeitintervalls dar. Das Ende wird hierbei implizit durch den Beginn der zeitlich nachfolgenden Ausprägung dieses Datensatzes gebildet.

Zeitintervall

Entscheidet sich der Datenbankdesigner hingegen für eine Zeitstempelung auf Grundlage eines Zeitintervalls, müssen die entsprechenden Tabellen im Rahmen der Gültigkeitszeit bzw. Transaktionszeit jeweils um zwei zeitliche Attribute ergänzt werden. Das eine Attribut stellt dabei den Beginn und das andere das Ende des Zeitintervalls dar. Hierbei gilt es zu entscheiden ob ein abgeschlossenes oder rechtsseitig halboffenes Zeitintervall Anwendung findet. Ein rechtsseitig halboffenes Intervall hat den Vorteil, dass bei der Überprüfung ob Zeitintervalle von zwei Tupeln lückenlos sind kein Chronon zum Ende aufaddiert werden muss.

Weiterhin ist in diesem Zusammenhang relevant, welcher Wert zur Darstellung eines unbestimmten Intervallendes gewählt wird. Da es hierfür im Rahmen von SQL keine konkreten Werte gibt, liegt diese Wahl in den Händen des Datenbankdesigners. Möglich sind hierbei ein selbst definierter Wert, NULL oder der größtmögliche Wert den der zeitliche Datentyp des entsprechenden Attributs annehmen kann. Die beiden ersten Alternativen sind dabei eher problematisch, wenn es beispielsweise darum geht Tupel zu selektieren die zu einem bestimmten Zeitpunkt gültig waren. Dafür wird es erforderlich die Intervallgrenzen der Tupel mit dem gewünschten Zeitpunkt zu vergleichen, wobei der Vergleich eines Datums mit NULL immer falsch bzw. unbekannt ausgehen würde, auch wenn diese Tupel zum gewünschten Zeitpunkt gültig waren.

Dies soll exemplarisch dargestellt werden. Hierfür ist nachfolgend eine entsprechende SQL-Anweisung formuliert. Diese soll sämtliche Datensätze aus der darunter folgenden Tabelle ausgeben die zum Zeitpunkt 2016-08-30 gültig waren. In dieser wird beschrieben über welchem Zeitraum Danny in Magdeburg gewohnt hat, wobei alle drei Datensätze mit einem anderen Intervallende versehen sind. Jeder dieser Datensätze ist zu diesem Zeitpunkt gültig, jedoch wird in Folge dieser Abfrage nur der

Datensatz mit dem Intervallende 9999-12-31 selektiert. Der Hintergrund hiervon ist, dass der Vergleich ob Max bzw. NULL größer als 2016-08-30 ist vom DBMS mit unbekannt bewertet wird. Aus diesem Grund wird in der Regel der größtmögliche Wert eines Zeitattributs zur Darstellung eines unbestimmten Intervallendes gesetzt.

```
SELECT *
FROM Tabelle
WHERE Beginn <= 2016-08-30
AND Ende > 2016-08-30
```

Name	Wohnort	Beginn	Ende
Danny	Magdeburg	1992-01-07	Max
Danny	Magdeburg	1992-01-07	NULL
Danny	Magdeburg	1992-01-07	9999-12-31

Tabelle 2.13: Datensätze mit drei alternativen Intervallenden

2.2.2.2 Primärschlüssel

Ein Objekt im Kontext temporaler Relationen besteht aus einer Reihe zeitlicher Ausprägungen. Aus diesem Grund wird in diesem Kontext von einem temporalen Objekt gesprochen. Um die Ausprägungen eines temporalen Objekts eindeutig identifizieren zu können existieren zwei Möglichkeiten: Die erste Möglichkeit besteht darin einen Surrogatschlüssel zu verwenden. Eine zweite Möglichkeit ist die zeitliche Dimension mit in den Primärschlüssel aufzunehmen. Dadurch besteht der Primärschlüssel jedes temporalen Objekts aus einem sog. Objekt-Schlüssel, welcher für ein Objekt immer gleichbleibend ist und jedes Objekt eindeutig identifizierbar macht sowie einem zeitlichen Bezug.

Wird die Gültigkeitszeit bzw. Transaktionszeit hierbei durch einen Zeitpunkt abgebildet, so müsste lediglich dieser zum Primärschlüssel hinzugefügt werden. Anders sieht es bei einer Abbildung durch ein Zeitintervall aus. Hierbei stehen der Beginn oder das Ende des Zeitintervalls zur Auswahl. Prinzipiell sind beide Alternativen gleichwertig. Insbesondere im Kontext der Transaktionszeit macht es jedoch Sinn, das Intervallende mit in den Primärschlüssel aufzunehmen, weil dort häufig nur die aktuelle Ausprägung eines Datensatzes von Interesse ist. Möglich ist jedoch auch stattdessen einen Surrogatschlüssel zu verwenden.

2.2.2.3 Integritätsbedingungen

Aus der Abbildung temporaler Konzepte in herkömmlichen relationalen Datenbanken erwachsen besondere Integritätsbedingungen, welche nur mit Hilfe von Anwendungslogik oder DBMS-Integrierter Funktionen wie z.B. Trigger oder Prozeduren, realisiert werden können.

Im Zusammenhang mit der Abbildung der temporalen Konzepte als Zeitintervall gilt prinzipiell zunächst, dass beim Einfügen eines jeden Datensatzes dessen Intervallende zeitlich nicht vor seinem Beginn liegen darf [Myr05]. Im Kontext der Transaktionszeit ist dies nicht problematisch, weil die Angabe der Intervallgrenzen vollkommen

eigenständig von der Anwendung übernommen wird. Dahingegen kann im Rahmen der Gültigkeitszeit der Nutzer die Intervallgrenzen bestimmen. Deswegen gilt es insbesondere hierbei die Einhaltung dieser Bedingung sicherzustellen.

Weiterhin dürfen sich hierbei die Zeitintervalle der Ausprägungen eines temporalen Objekts nicht überschneiden, weil dies sonst bedeuten würde, dass zwei Ausprägungen zur gleichen Zeit gültig bzw. aktuell waren [Myr05]. Auch diese Integritätsbedingung ist aus denselben Gründen für die Transaktionszeit nicht relevant, jedoch für die Gültigkeitszeit schon.

Über dies ergeben sich unabhängig davon ob die temporalen Konzepte als Zeitintervall oder Zeitpunkt dargestellt werden Integritätsbedingungen die eingehalten werden müssen um die referenzielle Integrität der temporalen Daten sicherzustellen. Dies ist der Fall, wenn sowohl die referenzierende, als auch die referenzierte Tabelle um temporale Konzepte ergänzt wurden. Hierbei müssen folgende drei Bedingungen zwingend eingehalten werden um die referenzielle Integrität im Kontext der Gültigkeitszeit bzw. Transaktionszeit zu gewährleisten [Myr05]:

1. Zu jedem auf ein temporales Objekt referenzierenden Datensatz muss es eine Ausprägung des referenzierten Objekts geben, dessen Intervallbeginn vor oder zur gleichen Zeit wie der Beginn des referenzierenden Datensatzes liegt.
2. Zu jedem auf ein temporales Objekt referenzierenden Datensatz muss es eine Ausprägung des referenzierten Objekts geben, dessen Intervallende nach oder zur gleichen Zeit wie das Ende des referenzierenden Datensatzes liegt.
3. Für jede Ausprägung eines referenzierten temporalen Objekts deren Intervallende in das eines referenzierenden Datensatzes fällt, muss eine zeitlich nachfolgende Ausprägung existieren.

3. Anforderungsanalyse

In diesem Kapitel werde ich die technischen Anforderungen an das digitale Archiv des VGBS analysieren und priorisieren.

Die Analyse der Anforderungen und die anschließende Priorisierung sind besonders wichtig, weil sie ermöglichen zu verstehen, was eine Archivierungslösung im Kontext der neu-entwickelten Verwaltungsanwendung des VGBS leisten sollte. Anhand der priorisierten Anforderungen lässt sich direkt die Qualität einer Archivierungslösung im Sinne des VGBS beurteilen. Mit Hilfe dieser können verschiedene Alternativen miteinander verglichen werden und dadurch die Lösung gefunden werden, welche die an sie gestellten Anforderungen am besten erfüllt.

Aus diesem Grund werde ich zunächst eine Ist-Analyse der Verwaltung des Vereins durchführen. Diese umfasst das Beschreiben des bisherigen Verwaltungssystems und die dazugehörige Archivierungsstrategie des VGBS sowie ein kurzer Überblick über die datenbankgestützte Verwaltungsanwendung, welche die Nachfolge dieses Systems antreten wird. Die Ist-Analyse ist wichtig zum Verständnis sowohl der im folgendem Schritt analysierten Anforderungen, als auch für die Vergabe der Prioritäten an diese. Überdies bildet sie die Basis der in Kapitel 4 vorgestellten datenbankinternen und datenbankexternen Methode.

Auf die Ist-Analyse aufbauend werde ich dann die technischen Anforderungen an die Archivierung der Vereinsdaten aus den Anforderungen des VGBS, den rechtlichen Forderungen sowie aus Fachliteratur ableiten.

Im Anschluss daran werde ich die ermittelten Anforderungen, mit Hilfe der AHP-Methode, nach dem Grad der Wichtigkeit für den Verein priorisieren. Bei der AHP-Methode werden alle Anforderungen paarweise miteinander verglichen und anhand des Grads an Wichtigkeit für den VGBS bewertet und priorisiert [Saa90].

3.1 Ist-Analyse

Dieser Abschnitt ist der Analyse des Ist-Zustandes der Verwaltung sowie Archivierung des VGBS gewidmet. Ist-Analyse bedeutet, dass ich in diesem Abschnitt erläutern werde wie das bisherige Verwaltungssystem des Vereins strukturiert ist und werde darauf aufbauend die Archivierung des VGBS schildern. Im Anschluss daran werde ich einen kurzen Überblick, über die neue Verwaltungsanwendung des Vereins geben.

3.1.1 Das bisherige Verwaltungssystem

Der Verein für Gesundheit, Bewegung und Sport ist ein gemeinnütziger Sportverein, welcher sich seit 2004 dem Rehabilitations- sowie Gesundheitssport verschrieben hat. Hierfür bietet der VGBS ein breites Spektrum an Sportangeboten für seine Mitglieder, aber auch für Personen denen Sport ärztlich verschrieben wurde genannt Nicht-Mitglieder, sowie für jene die an einer der Vielzahl von Kursen des Vereins teilnehmen wollen.

Zur Verwaltung der hieraus resultierenden Daten nutzt der VGBS ein System, das im Kern aus drei Microsoft Excel-Dateien besteht. Wie in [Abbildung 3.1](#) zu sehen, liegen diese auf einer externen Serverfestplatte, welche sich wiederum in den Räumlichkeiten des Vereins befindet. Dabei werden in diesen Dateien so gut wie alle Daten, die im operativen Betrieb des Vereins anfallen, gespeichert und verwaltet. Dies umfasst relevante Daten der Mitglieder, Nicht-Mitglieder und Kursteilnehmer des Vereins wie z.B. Adressen, Telefonnummern und Medizinische Daten darunter einzunehmende Medikamente und Krankheitsbilder, aber auch gezahlte Beiträge. Weiterhin werden in diesen Dateien die Übungsleiter sowie Sportangebote des VGBS verwaltet. Dies bedeutet, dass relevante Daten von Übungsleitern wie beispielsweise Namen bzw. Anschriften gespeichert, sowie Anwesenheitslisten geführt und Trainingspläne aufgestellt werden.

Um Zugriff auf diese Daten zu erhalten benötigen die Mitarbeiter des VGBS ein Passwort, mit dessen Hilfe sie sich an einem Rechner des Vereins anmelden können. Von dort aus ist es jedem Mitarbeiter möglich, auf die drei Excel-Dateien und somit auch auf sämtliche oben beschriebenen Daten zuzugreifen.

3.1.1.1 Archivierung

Um die oben beschriebenen Daten eines jeweiligen Jahres zu sichern, werden am Beginn bzw. Ende eines jeden Jahres die 3 Excel-Dateien vom Verein archiviert. Wie dies abläuft kann in [Abbildung 3.2](#) betrachtet werden. Zunächst werden die Excel-Dateien dupliziert, woraus zwei Versionen jeder Datei resultieren. Die eine Version ist dabei zur Archivierung bestimmt, wohingegen die Andere weiterhin operativ genutzt wird.

Die zu archivierenden Dateien werden hierzu an einen Speicherort auf der Serverfestplatte geschoben, welcher separat von den operativen Dateien ist. Dort lagernd, werden diese nicht weiter operativ genutzt, sodass die in ihnen gespeicherten Daten in der Form erhalten bleiben, wie sie am Ende eines Jahres waren. Damit bildet jede archivierte Datei sozusagen den letzten Schnappschuss der Vereinsdaten eines Jahres ab.

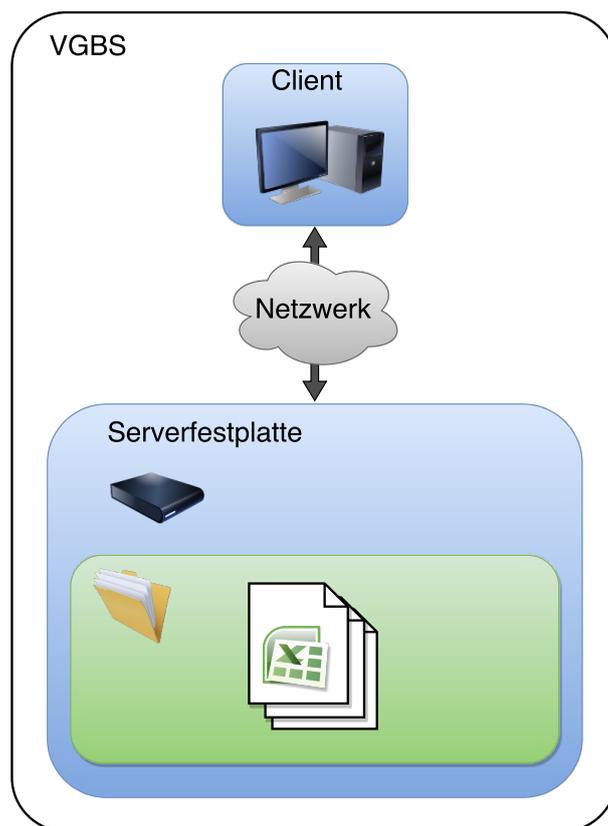


Abbildung 3.1: Das bisherige Verwaltungssystem

Darüber hinaus werden die Excel-Dateien, welche weiterhin operativ genutzt werden, bereinigt. Dies bedeutet, dass sämtliche jahresbezogenen Daten, wie beispielsweise über das Jahr gezahlte Beiträge oder geführte Anwesenheitslisten, aus diesen Dateien gelöscht werden. Damit stellt der VGBS sicher, dass die in diesen Dateien gespeicherten Daten wirklich nur die des jeweiligen Jahres sind. Das heißt, dass sich darunter keine Daten aus dem letzten Jahr befinden, die nicht für das aktuelle Jahr relevant sind.

Die Archivierung im Kontext des VGBS erfüllt dabei verschiedene Zwecke: Zum einen ist der Verein dadurch in der Lage nachzuvollziehen, welche relevanten Informationen über seine Mitglieder, Nicht-Mitglieder und Kursteilnehmer pro Jahr gespeichert wurden. Diese Informationen dienen dem Verein in erster Linie für organisatorische Zwecke.

Auf der anderen Seite dient die Archivierung der Excel-Dateien insbesondere dem Zweck die darin gespeicherten Daten auf gesetzliche Nachfrage hin bereitstellen zu können. Dies kann auf Grund eines Rechtsstreits der Fall sein, aber auch im Falle einer steuerrechtlichen Außenprüfung, bei der die Prüfung aller steuerlich relevanten Sachverhalte durchgeführt wird. Dies schließt die in den archivierten Excel-Dateien gespeicherten Daten mit ein. In diesem Zusammenhang gibt es verschiedene Zeiträume innerhalb deren die jährlichen Daten des VGBS archiviert werden müssen. Diese werden als Aufbewahrungsfristen bezeichnet.

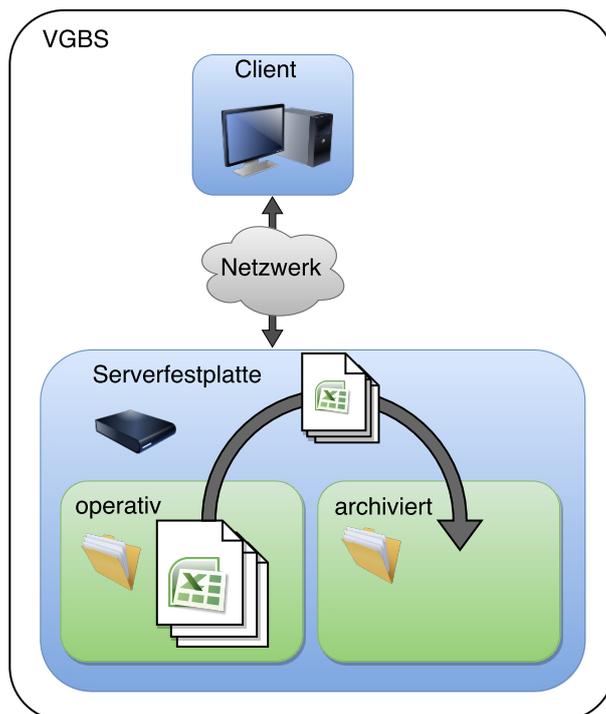


Abbildung 3.2: Archivierung im Kontext des bisherigen Verwaltungssystems.

3.1.1.2 Aufbewahrungsfristen

Im Kontext der Archivierung des VGBS gelten aus rechtlicher Sicht verschiedene Zeiträume sog. Aufbewahrungsfristen innerhalb denen die Daten des VGBS archiviert werden müssen und somit vom Gesetzgeber eingesehen werden können.

Laut §147 AO gelten Aufbewahrungsfristen von 10 Jahren für Bücher, Aufzeichnungen, Inventare, Jahresabschlüsse, Buchungsbelege und Lageberichte. Außerdem für die Eröffnungsbilanz sowie die zu ihrem Verständnis erforderlichen Arbeitsanweisungen und sonstigen Organisationsunterlagen. Aufbewahrungsfristen von 6 Jahren gelten für empfangene und Wiedergaben von abgesandten Handels- oder Geschäftsbriefen sowie für sonstige Unterlagen, soweit sie für die Besteuerung von Bedeutung sind.

Die archivierten Excel-Dateien des VGBS beinhalten hierbei in erster Linie Bücher und Aufzeichnungen beispielsweise in Form von gezahlten Beiträgen sowie sonstige Unterlagen wie z.B. Personalunterlagen. Um im Rahmen dessen den hierfür erforderlichen Aufbewahrungsfristen gerecht zu werden, archiviert der Verein seine Excel-Dateien jeweils 10 Jahre, bevor diese gelöscht werden. Damit stellt der VGBS sicher, dass wirklich alle Daten über ihre jeweilige Aufbewahrungsfrist archiviert werden.

3.1.1.3 Anpassungen

Im Rahmen der im vorhergehenden Abschnitt erläuterten Aufbewahrungsfristen hat sich in der 12 Jährigen Geschichte des VGBS die Erfordernis aufgetan archivierte Excel-Dateien nachträglich anzupassen. Das heißt neue archivierte Daten werden verfasst und bestehende bearbeitet bzw. gelöscht. Diese Maßnahmen nimmt der

VGBS vor, um die darin enthaltenen Daten an die, aus der Sicht des Vereins, reellen Gegebenheiten anzupassen. So kann es vorkommen, dass ein Mitglied einen Beitrag aus dem vorhergehendem Jahr im aktuellen nachzahlt. Da die operativen Excel-Dateien nur Daten aus dem aktuellen Jahr beinhalten, trägt der Verein diesen Sachverhalt in der entsprechenden archivierten Excel-Datei nach. Damit wäre im Falle einer gesetzlichen Auswertung der archivierten Vereinsdaten erkennbar, dass dieser Beitrag vollständig gezahlt wurde.

3.1.2 Die Verwaltungsanwendung

Nach den vielen Jahren in denen der VGBS das in [Abschnitt 3.1.1](#) beschriebene Verwaltungssystem nutzte, stellt dieses sich in der heutigen Zeit nicht mehr als adäquat heraus, um die Bedürfnisse des Vereins weiterhin vollständig zu erfüllen. So ermöglichte dieses System stets nur einem Mitarbeiter gleichzeitig den Zugriff auf eine Excel-Datei. In Anbetracht der vielen oft gleichzeitig stattfindenden und zu verwaltenden Sportgruppen ist dies eine Belastung für den VGBS. Eine weitere Belastung stellen in diesem Zusammenhang diverse Abstürze der Excel-Dateien sowie deren Trägheit, welche aus der Masse der in ihnen gespeicherten Daten resultiert, dar.

Um diesen und weiteren Problemen im Rahmen der Verwaltung des VGBS Herr zu werden, gab der Verein in Auftrag, eine auf Webtechnologie gestützte Verwaltungsanwendung zu entwickeln. Mit deren Hilfe sollen dessen Vereinsdaten, an die aktuellen Bedürfnisse des VGBS angepasst, datenbankgestützt verwaltet werden können. In [Abbildung 3.3](#) ist eine Übersicht der an der Verwaltungsanwendung beteiligten Komponenten zu sehen.

Dabei besteht die Struktur der Verwaltungsanwendung im wesentlichen aus einer MySQL-Datenbank und darauf aufbauend der eigentlichen Verwaltungsanwendung. Beide befinden sich auf einem Server im Internet. Hierbei werden zukünftig die Vereinsdaten in der MySQL-Datenbank gespeichert. Da MySQL ein relationales DBMS ist, befinden sich diese Daten in normalisierten Tabellen verteilt in der Datenbank. Verwaltet werden diese Daten mittels der Verwaltungsanwendung, welche SQL-Anweisungen nutzt um Daten anzufordern oder zu verändern. Möchte ein Mitarbeiter des VGBS Zugriff auf diese Daten erhalten, so muss er über einen Internet-Browser auf die Verwaltungsanwendung zugreifen. Diese stellt ihm daraufhin eine Benutzeroberfläche zum Verwalten der Daten bereit.

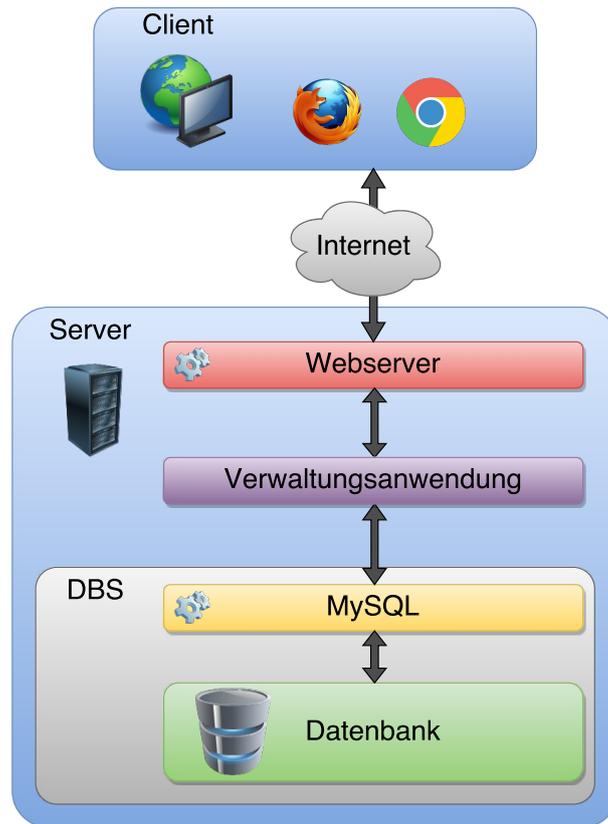


Abbildung 3.3: Die neu-entwickelte Verwaltungsanwendung

3.2 Anforderungen

In diesem Abschnitt werde ich die technischen Anforderungen an die digitale Archivierung des Vereins analysieren. Diese lassen sich grundlegend in zwei Kategorien einordnen: Zum einen in Allgemeine Anforderungen. Diese sollten prinzipiell von jedem Archiv, das relationale Daten speichert und verwaltet, erfüllt werden. Auf der Seite lassen sie sich in jene Anforderungen einordnen, welche konkret vom VGBS gestellt werden. Aus diesem Grund ist der nachfolgende Abschnitt auch auf diese Weise gegliedert.

Einzelne Anforderungen werden dabei in folgender Struktur erläutert: Zunächst wird eine Anforderung in einem prägnanten Satz zusammengefasst. Dieser befindet sich in kursiver Schrift in einem blau-umrahmten Kasten. Hierbei werden einzelne Wörter, welche die beschriebene Anforderung charakterisieren dick gedruckt. Hier ein Beispiel:

*Ich bin ein **Beispiel** für eine Anforderung.*

Unter diesem Kasten wird falls notwendig eine Erklärung zu dieser Anforderung folgen. Auf diese folgt wiederum die Beschreibung der Quelle der erläuterten Anforderung.

3.2.1 (A) Allgemeine Anforderungen

Dieser Abschnitt widmet sich der Analyse der allgemeinen technischen Anforderungen, die an das digitale Archiv des Vereins gestellt werden. Diese werden im Kontext dieser Arbeit stets mit einem großen A, gefolgt von einer eindeutigen Nummer, bezeichnet. Mit Hilfe dieses Index lässt sich jede Anforderung eindeutig identifizieren.

Ferner leiten sich diese Anforderungen aus dem rechtlichen Kontext, sowie Fachliteratur ab und sollten somit von jedem digitalen Archiv relationaler Daten erfüllt werden. Hierbei existieren zum einen verschiedene Rechtstexte die sich mit der Aufbewahrung von steuerlich relevanten Daten beschäftigen. Zu diesen gehören u.a. die Abgabenordnung und das Handelsgesetzbuch, wobei in keinem der beiden der Begriff Archivierung Erwähnung findet.

Vielmehr lassen sich aus diesen Rechtstexten Anforderungen an die revisionssichere digitale Archivierung ableiten. So werden beispielsweise in [dL14, uIeV09] die grundlegenden Charakteristika der revisionssicheren Archivierung von diesen abgeleitet. In diesem Kontext wäre auch Kampffmeyer erwähnt, welcher ebenfalls diese Rechtstexte aufgreift, jedoch noch weitere Anforderungen an ein digitales Archiv definiert. Kampffmeyer ist ein deutscher Autor und Unternehmensberater, der sich schon seit einigen Jahren intensiv mit dem Thema der revisionssicheren elektronischen Archivierung auseinandersetzt. In diesem Bereich hat er bereits an vielen Publikationen, darunter [KR97, Kam06a, Kam06b], mitgewirkt.

Zum anderen hat sich Herbst in [HKS95] insbesondere mit dem Thema der Datenbankarchivierung beschäftigt. Dabei definiert er sowohl grundlegende Anforderungen an ein digitales Archiv, als auch Anforderungen an das sog. anwendungsorientierte DB-Archivieren. Anwendungsorientiertes DB-Archivieren ist ein Dienst der vom DBS bereitgestellt wird, um Daten zu archivieren die im DBS gespeichert sind [HKS95].

Über dies existieren Modelle die den strukturellen Aufbau eines digitalen Archivs aus einer Top-Down-Perspektive beschreiben. Dies bedeutet, dass die Kernfunktionen eines solchen Archivs in abstrakter Weise definiert werden, ohne Rücksicht auf bestimmte Datenformate oder Technologien. Zu diesen gehört beispielsweise das OAIS-Modell. Das Open Archival Information System kurz OAIS ist ein dynamisches erweiterbares Referenzmodell für digitale Archive [ISO12]. Im Gegensatz dazu werden die Anforderungen an ein digitales Archiv in [RRL⁺05] aus einer Bottom-Up-Perspektive, also aus technologischer Sicht, betrachtet.

Die von diesen Autoren beschriebenen Anforderungen lassen sich mal mehr und mal weniger auch auf die technischen Anforderungen an ein revisionssicheres digitales Archiv relationaler Daten übertragen. Somit finden sich diese teilweise in ihrer ursprünglichen Form oder auch an die Datenbankarchivierung angepasst in dieser Arbeit wieder.

3.2.1.1 (A1) Aufnahmegranularität

*Der Nutzer sollte die Möglichkeit besitzen so **feingranular zu archivieren**, wie es seinen Anforderungen entspricht.*

Feingranular bedeutet, dass Informationsobjekte jeder Größenordnung archiviert werden können. Das heißt ein zu archivierendes Informationsobjekt könnte aus einzelnen Datensätzen oder einem ganzen Verbund von diesen bestehen.

Die Forderung nach Granularität im Kontext des Datenbankarchivierens wird in [HKS95] von Herbst aufgeworfen. Dieser definiert Granularität als eine Anforderung an das eingangs beschriebene sog. anwendungsorientierte DB-Archivieren. Prinzipiell lässt sich diese Anforderung aber auf jedes Archiv übertragen, das Daten aus einer relationalen Datenbank speichert und verwaltet. Denn insbesondere in diesem Kontext können einzelne zu archivierende Informationsobjekte aus Attributen, Tupeln, ganzen Tabellen, Views oder sogar der gesamten Datenbank bestehen. Dabei ist es nicht immer notwendig und sinnvoll nur eine dieser Formen zu archivieren.

3.2.1.2 (A2) Direkter Zugriff

*Das Archiv sollte dem Nutzer **direkten Zugriff auf einzelne Informationsobjekte** erlauben.*

Diese Anforderung wird u.a. von Kampffmeyer in [Kam06a] an ein elektronisches Archiv gestellt. Wie bereits im Abschnitt Aufnahmegranularität erläutert, können Informationsobjekte im Kontext relationaler Datenbanken von verschiedener Struktur sein. Wobei ein archiviertes Informationsobjekt wie zum Beispiel eine Tabelle noch weitere Informationsobjekte enthalten kann. Im Fall der Tabelle wären dies einzelne Tupel oder Attribute. Dem Nutzer sollte es dabei möglich sein direkt auf einzelne dieser Strukturelemente eines Informationsobjekts zugreifen zu können d.h. ohne zunächst das gesamte Informationsobjekt wiederherstellen zu müssen.

3.2.1.3 (A3) Verlustfreiheit

*Das Archiv muss Informationsobjekte **verlustfrei wiederherstellen** können.*

Verlustfrei bedeutet in diesem Kontext, dass archivierte Daten exakt in der Form wiederhergestellt werden, in der sie ursprünglich gespeichert worden.

Konkret lässt sich diese Forderung u.a. in [uIeV09] und in [HKS95] wiederfinden. In [uIeV09] wird der verlustfreie Zugriff zwar im Kontext digitaler Dokumente in Form von gescannten physischen Dokumenten erwähnt, grundlegend lässt sich dies jedoch natürlich auf jedes archivierte Objekt übertragen. Dabei heißt es in [uIeV09], dass archivierte Dokumente zu jedem Zeitpunkt ihrer Aufbewahrung mit ihrem Original übereinstimmen müssen.

3.2.1.4 (A4) Benutzerveranlassung

*Das Archiv muss dem Nutzer die Möglichkeit bereitstellen, den **Archivierungsvorgang entweder manuell oder ereignisgesteuert zu veranlassen.***

Benutzerveranlassung ist ebenfalls eine Anforderung die Herbst in [HKS95] an das anwendungsorientierte DB-Archivieren stellt.

Diese Anforderung ist laut Herbst wichtig, weil nur dem Nutzer bekannt ist, welche Daten, wann zu archivieren sind. Aus diesem Grund sollte eine Archivierung von Daten nicht ausschließlich durch implizite Kriterien wie zum Beispiel deren Alter oder Speichervolumen ausgelöst, sondern direkt vom Nutzer gesteuert werden.

3.2.1.5 (A5) Ordnungskriterien

*Das Archiv sollte dem Nutzer Funktionen bieten, mit deren Hilfe er ausgewählte **Ordnungskriterien** auf archivierte Daten anwenden kann.*

Ordnungskriterien könnten hierbei die Zeit sein, also das Alter der archivierten Daten, oder eine alphabetische Sortierung nach Namen.

Diese Forderung resultiert aus [dL14] und wurde darin aus §146 Absatz 1 AO abgeleitet. In [dL14] wird aus diesem Paragraph gefolgert, dass Bücher und Aufzeichnungen nach festgelegten Ordnungskriterien zu führen sind. Da Archive unter anderen auch diese Daten enthalten trifft diese Aussage auch auf sie zu.

Weiterhin greift auch Kampffmeyer die Forderung nach Ordnungskriterien eines Archivs auf. In [Kam06b] bezeichnet er das aufstellen von Ordnungskriterien als „*eine der Hauptaufgaben bei der Planung und Einführung eines Archivsystems*“.

3.2.1.6 (A6) Indexierung

*Jedem archivierten Informationsobjekt muss ein **eindeutiger Index** zugeteilt werden.*

Unter eindeutig ist zu verstehen, dass es keine zwei im Archiv gespeicherten Informationsobjekte gibt, die den selben Index besitzen.

In [Kam06a] ist die Indexierung eine Forderung von Kampffmeyer die an ein elektronisches Archiv gestellt wird. Auch in [dL14] wird nach der Zuteilung eindeutiger Indizes zu einzelnen archivierten Informationsobjekten verlangt. Dabei heißt es ergänzend, dass diese Zuteilung über den gesamten Zeitraum der Aufbewahrung der indizierten Informationsobjekte erhalten bleiben muss, sowie das diese unter ihren zugeordneten Indizes verwaltet werden sollen.

3.2.1.7 (A7) Unabhängigkeit

*Das Archiv muss seine Daten **unabhängig** von ihrem Erzeugersystem **lagern, verwalten und bereitstellen** können.*

Als Erzeugersystem verstehen sich die Programme mit deren Hilfe die archivierten Daten entstanden sind, aber auch die Repräsentation in der sie entstanden sind.

Dazu gehört beispielsweise die Verwaltungsanwendung des VGBS, aber auch das Betriebssystem und das Dateiformat, in dem die im Archiv gespeicherten Daten ursprünglich erzeugt wurden.

Ferner stellt nur die Unabhängigkeit der archivierten Daten von ihrem Erzeugersystem sicher, dass diese auch in vielen Jahren noch verlustfrei wiederhergestellt werden können.

Die Unabhängigkeit eines Archivs vom Erzeugersystem der in ihm gespeicherten Daten ist eine Forderung die unter verschiedenen Aspekten von vielen Autoren Erwähnung findet. Für Herbst ist dies in [HKS95] eine qualitative Anforderung an ein elektronisches Archiv. Laut Herbst könnte Unabhängigkeit eines Archivs durch neutrale genormte Schnittstellen innerhalb des Archivsystems und nach außen erreicht werden. Weiterhin führt Herbst aus, dass im Kontext des anwendungsorientierten DB-Archivieren, das DBS dedizierte Operationen besitzen sollte mit deren Hilfe dauerhaft auf die Daten des Archivs zugegriffen werden kann. Nur so könnte laut Herbst die Unabhängigkeit des archivierten Datenbestandes von seinem Erzeugersystem gewährleistet sein. Auch [Kam06b] erwähnt dies als Anforderung an ein elektronisches Archiv. Weitere Beispiele wären das speichern der archivierten Daten in langlebige, verbreitete und nach Möglichkeit für Menschen lesbare Dateiformate. Beispiele hierfür sind CSV, XML oder auch das eigens zur Datenbankarchivierung entwickelte XML-Derivat DBML (Database Markup language) [JLRH02].

3.2.1.8 (A8) Agilität

*Das Archiv sollte **agil** gegenüber organisatorischen Änderungen der archivierenden Institution sein.*

Organisatorische Änderungen im Rahmen des VGBS sind beispielsweise, dass dieser zukünftig auch DVDs mit Sportübungen für zu Hause verkauft und dafür keine Sportangebote mehr für Nicht-Mitglieder anbietet. Die Art dieser DVDs und die Einnahmen, die aus ihnen resultieren, würden sich dann früher oder später auch im Archiv wiederfinden. Die Daten von Sportgruppen aus dem Bereich der Nicht-Mitglieder würden an der Stelle nicht archiviert werden können, weil sie nicht existieren. In diesem Fall sollte Archiv trotzdem auch die archivierten Daten von vor dieser organisatorischen Veränderung wiederherstellen können.

Die Forderung nach Agilität leitet sich aus der Natur relationaler Datenbanken ab. Hierbei können sich solche Veränderungen direkt auf das Datenbankschema auswirken. Würde dieses jedoch genutzt werden um die archivierten Daten wieder zugänglich zu machen, so käme es zu Konflikten auf Grund fehlender oder zusätzlicher Schemaelementen wie z.B. Tabellen oder Attribute.

3.2.1.9 (A9) Metadaten

*Archivierte Daten sollten um **Metadaten** ergänzt werden.*

Metadaten werden zum identifizieren, verwalten, lokalisieren und interpretieren von archivierten Informationsobjekten genutzt [vBD06]. Hierzu beschreiben Metadaten

den Inhalt eines Informationsobjekts, seinen Erzeuger bzw. Erzeugersystem, seine Repräsentationsform, sowie seine Beziehung zu anderen Informationsobjekten.

Metadaten finden besonders im OAIS-Modell Anwendung. Hierbei werden jedem sog. Information Package Metadaten zugeteilt [ISO12]. Ein Information Package ist ein Informationsobjekt im Rahmen des OAIS-Modells.

3.2.1.10 (A10) Zugriffsschutz

*Das Archiv sollte über einen **Zugriffsschutz** verfügen, welcher **nur berechtigten Nutzern Zugriff** auf die archivierten Daten gestattet.*

Die Forderung nach einem Zugriffsschutz im digitalen Archiv ist in erster Linie aus Datenschutzrechtlichen Gründen erforderlich. Archivierte Daten umfassen häufig vertrauliche Personenbezogene Daten. Im Kontext des VGBS sind dies u.a. Anschriften, Telefonnummern, Krankheitsbilder und eingenommene Medikamente (siehe Abschnitt 3.1.1 „Das bisherige Verwaltungssystem“). Ausschließlich qualifiziertes befugtes Personal sollte Einsicht in diese Daten bekommen.

Diese Anforderung findet sich in verschiedenen rechtlichen Quellen wieder, darunter [uIeV09, KR97]. Auch in Fachliteratur wie beispielsweise in [HKS95], sowie in Modellen zum strukturellen funktionalen Aufbau eines Archivs, wie zum Beispiel dem OAIS-Modell, findet sich diese Anforderung wieder [ISO12].

3.2.1.11 (A11) Unveränderbarkeit

*Die archivierten Daten müssen über ihre gesamte Aufbewahrungsfrist **unveränderbar archiviert** werden.*

Dies bedeutet, dass jene Daten nicht in einer Form verändert werden dürfen, die es unmöglich macht ihre ursprüngliche Form festzustellen. Außerdem muss feststellbar sein, ob diese Veränderung ursprünglich oder nachträglich stattfand. Damit bleibt die Authentizität der gespeicherten Daten auch bei einer nachträglichen Modifikation erhalten.

Diese Anforderung resultiert direkt aus §146 Absatz 4 AO und §239 Absatz 3 HGB und wurde konkret nochmal in [dL14] festgehalten.

3.2.1.12 (A12) Angemessenheit

*Archivierte Daten müssen in **angemessener Zeit wiederhergefunden** und ggf. **reproduziert** werden können.*

Welche Zeit als angemessen angesehen wird, hängt dabei vom Stand der Technik, sowie vom gegebenen Kontext ab.

Dies ist eine Anforderung an den Archivzugriff, welche sich indirekt aus § 145 Absatz 1 AO ableiten lässt und konkret in [uIeV09, HKS95] geschrieben steht.

3.2.2 (V) VGBS-Anforderungen

In diesem Abschnitt werde ich Anforderungen erläutern, die direkt vom VGBS gestellt werden. Zur Unterscheidung von den allgemeinen Anforderungen, werden diese mit einem großen V bezeichnet. Dieses wird wieder gefolgt von einer Nummer, welche die Eindeutigkeit der Bezeichnungen sicherstellt.

Die Struktur des bisherige Archivierungssystems des VGBS führte zu einem dazu, dass jeweils der gesamte Datenbestand eines Jahres archiviert wurde (siehe [Abschnitt 3.1.1](#) „Das bisherige Verwaltungssystem“). Wie bereits erläutert blickte der VGBS beim Zugriff auf eine archivierte Excel-Datei jeweils nur auf den letzten Zustand seiner Vereinsdaten aus dem entsprechenden Jahr. Da sich der Verein an dieses Verhalten im Laufe seiner 12-jährigen Geschichte gewöhnt und dementsprechend seine organisatorischen Abläufe daran ausgerichtet hat, möchte der VGBS dies auch in seiner Verwaltungsanwendung weiterhin beibehalten.

Andererseits ermöglichte das bisherige Archivierungssystem des VGBS auch das umfassende Modifizieren seiner archivierten Vereinsdaten (siehe [Abschnitt 3.1.1.3](#) „Anpassungen“). Dafür können archivierte Excel-Dateien geöffnet werden und dort neue Datensätze verfasst und bestehende Datensätze bearbeitet oder gelöscht werden. Diese Möglichkeiten hat der Verein genutzt, um beispielsweise nachgezahlte Beiträge mit in das Archiv aufzunehmen oder eine archivierte Sportgruppenzugehörigkeit einer Person nachträglich zu löschen, weil diese doch nicht an der vermerkten Sportgruppe teilnehmen möchte.

Obwohl diese Anforderungen scheinbar gegensätzlich zur rechtlich vorgeschriebenen Unveränderbarkeit des archivierten Datenbestandes ist, möchte der VGBS sie auch in seinem digitalen Archiv im Kontext seiner datenbankgestützten Verwaltungsanwendung (siehe [Abschnitt 3.1.2](#) „Die Verwaltungsanwendung“) umgesetzt haben. Dies bedeutet, dass im Kontext der Verwaltungsanwendung nur Archivierungsmethoden in Frage kommen, welche die Anforderungen des Vereins und zugleich die Unveränderbarkeit des archivierten Datenbestandes erfüllen können. Dies wäre nur der Fall, wenn eine Archivierungsmethode die Modifikationen an den archivierten Daten protokolliert, sodass der ursprüngliche Zustand auch nach einer Veränderung dieser noch im Archiv gespeichert ist.

3.2.2.1 (V1) Jährlicher Schnappschuss

*Das Archiv soll den **letzten Zustand der Vereinsdaten pro Jahr** archivieren.*

Diese Forderung wird vom VGBS gestellt, um die Struktur des bisherigen Archivierungssystems des Vereins beizubehalten. Dabei stellt sie eine Präzisierung der Aufnahmegranularität sowie der Benutzerveranlassung dar. Die Aufnahmegranularität wurde in sofern präzisiert, dass diese auf den gesamten Datenbestand festgelegt wurde. Die Benutzerveranlassung wurde ebenfalls darauf begrenzt jährlich den Datenbestand zu archivieren, anstatt dem Nutzer die Wahl zu überlassen, wann dies geschehen soll.

3.2.2.2 (V2) Verfassen

*Das Archiv muss es dem VGBS ermöglichen **Daten in einem archivierten Dateibestand zu verfassen**.*

3.2.2.3 (V3) Bearbeiten

*Es muss dem Verein möglich sein, **Daten aus einem archivierten Datenbestand zu bearbeiten.***

3.2.2.4 (V4) Löschen

*Daten aus einem archivierten Datenbestand sollten vom VGBS, noch **vor dem Ablauf ihrer Aufbewahrungsfrist, gelöscht werden können.***

Die Aufbewahrungspflicht für der Bücher und Aufzeichnungen des VGBS beträgt 10 Jahre (siehe [Abschnitt 3.1.1.2 „Aufbewahrungsfristen“](#)). Erst nach Ablauf dieser Frist dürfen diese aus rechtlicher Sicht aus dem Archiv entfernt werden. Der Verein möchte jedoch die Möglichkeit besitzen seine archivierten Daten, unabhängig vom Inhalt, vor Ablauf dieser Frist löschen zu können.

3.3 Priorisierung

Dieser Abschnitt ist der Priorisierung der, im vorhergehenden Abschnitt abgeleiteten, Anforderungen gewidmet. Diese erfolgt anhand der Wichtigkeit dieser Anforderungen für die Archivierung des VGBS im Kontext seiner Verwaltungsanwendung (siehe [Abschnitt 3.1.2 „Die Verwaltungsanwendung“](#)). Hierzu wird der Analytic Hierarchy Process (AHP) verwendet. Näheres zu dieser Methode ist in [Abschnitt 2.1 „Analytic Hierarchy Process“](#) zu finden.

Der Ablauf des AHP umfasst folgende Schritte: Die Definition des Entscheidungsproblems, Konstruktion der Hierarchie, Vergleich und Bewertung, sowie Gewichtung und Lösung. Jeden diese Schritte werde ich im Folgendem ausführen, um letztendlich jede Anforderung in Bezug auf die restlichen zu priorisieren.

3.3.1 Definition des Entscheidungsproblems

Das Entscheidungsproblem dieser Arbeit besteht darin, eine geeignete Archivierungslösung für die Verwaltungsanwendung des VGBS auszuwählen. Die hierfür erforderlichen Kriterien wurden bereits im vorhergehenden [Abschnitt 3.2](#) identifiziert. Diese liegen in Form der Anforderungen vor, die an das digitale Archiv des VGBS gestellt werden.

3.3.2 Konstruktion der Hierarchie

Daraus folgend ergibt sich in diesem Schritt die in [Abbildung 3.4](#) illustrierte Hierarchie. Die Wurzel bildet hierbei das übergeordnete Ziel eine Archivierungsmethode für die relationalen Vereinsdaten auszuwählen. Der Wurzel untergeordnet sind alle im letzten Abschnitt identifizierten Anforderungen. Auf die *Aufnahmegranularität* und *Benutzervernalassung* wird dabei verzichtet, weil diese, wie in [Abschnitt 3.2.2](#) erwähnt, mit der Anforderung *Jährlicher Schnappschuss* schon in präzisierter Form vorliegen. Ferner werden die Ergebnisse des paarweisen Vergleichs der Anforderungen bereits in [Abbildung 3.4](#) in grau angedeutet.

Weiterhin existieren für diesen Anwendungsfall keine untergeordneten Subkriterien, weshalb die nächste Ebene durch die Alternativen gebildet wird. Die beiden Alternativen sind die datenbankinterne sowie datenbankexterne Methode. Der Vergleich dieser Archivierungsmethoden bezüglich der Anforderungen findet jedoch erst im späteren Verlauf dieser Arbeit statt, weil es zunächst erforderlich wird diese vorzustellen. Aus diesem Grund werden sie in [Abbildung 3.4](#) ebenfalls nur in grau angedeutet.

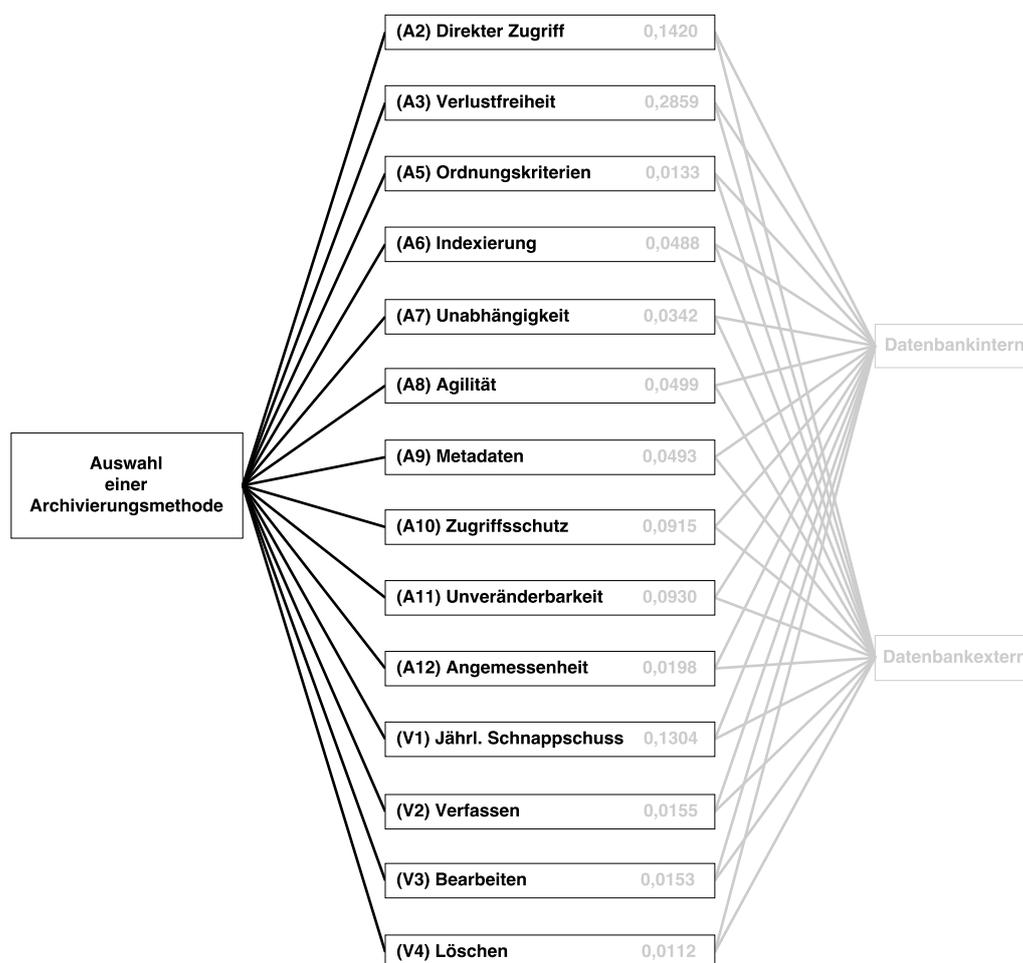


Abbildung 3.4: Hierarchie

3.3.3 Vergleich und Bewertung

Mit Hilfe der im letzten Schritt erstellten Hierarchie können jetzt paarweise die Elemente einer Ebene in Bezug auf das Element in der nächst höheren verglichen und bewertet werden. Der Vorteil dieser paarweisen Vergleiche ist, dass dadurch die AHP-Methode ein sehr robustes Verfahren ist, um Entscheidungsprobleme zu lösen. Dadurch ist sie im Vergleich zu anderen Methoden zur Priorisierung von Anforderungen wie z.B. MoSCoW, sehr aufwändig. Bei einer Anzahl n zu vergleichenden Elementen müssen im Rahmen des AHP $n * (n - 1) / 2$ Vergleiche durchgeführt werden. Somit sind es beim Vergleich der 14 in [Abbildung 3.4](#) dargestellten Anforderungen 91 paarweise Vergleiche die durchgeführt werden müssen. Für die weiteren Betrachtungen und die Auswahl einer Archivierungslösung ist die Bewertung der

Anforderungen resultierend aus den Vergleichen ausschlaggebend. Im Folgenden gebe ich deswegen nur einen Überblick über das Ergebnis der Gewichtung. Dieser ist in [Tabelle 3.1](#) illustriert. Die detaillierte Diskussion kann im Anhang A in [Abschnitt A.1](#) nachgeschlagen werden.

A vs. B	Besser	Bew.	A vs. B	Besser	Bew.	A vs. B	Besser	Bew.
A2 vs. A3	A3	9	A5 vs. A12	A5	4	A8 vs. V4	A8	5
A2 vs. A5	A2	8	A5 vs. V1	V1	8	A9 vs. A10	A10	7
A2 vs. A6	A2	8	A5 vs. V2	A5	3	A9 vs. A11	A11	8
A2 vs. A7	A2	7	A5 vs. V3	A5	3	A9 vs. A12	A9	7
A2 vs. A8	A2	7	A5 vs. V4	A5	3	A9 vs. V1	V1	7
A2 vs. A9	A2	7	A6 vs. A7	A6	4	A9 vs. V2	A9	7
A2 vs. A10	A2	7	A6 vs. A8	A8	3	A9 vs. V3	A9	7
A2 vs. A11	A2	7	A6 vs. A9	A9	2	A9 vs. V4	A9	7
A2 vs. A12	A2	8	A6 vs. A10	A10	6	A10 vs. A11	A11	2
A2 vs. V1	Beide	1	A6 vs. A11	A11	6	A10 vs. A12	A10	6
A2 vs. V2	A2	7	A6 vs. A12	A6	7	A10 vs. V1	V1	6
A2 vs. V3	A2	9	A6 vs. V1	V1	8	A10 vs. V2	A10	9
A2 vs. V4	A2	9	A6 vs. V2	A6	8	A10 vs. V3	A10	9
A3 vs. A5	A3	9	A6 vs. V3	A6	8	A10 vs. V4	A10	9
A3 vs. A6	A3	9	A6 vs. V4	A6	8	A11 vs. A12	A11	7
A3 vs. A7	A3	9	A7 vs. A8	A8	4	A11 vs. V1	V1	7
A3 vs. A8	A3	9	A7 vs. A9	A9	4	A11 vs. V2	A11	9
A3 vs. A9	A3	9	A7 vs. A10	A10	7	A11 vs. V3	A11	9
A3 vs. A10	A3	9	A7 vs. A11	A11	7	A11 vs. V4	A11	9
A3 vs. A11	A3	9	A7 vs. A12	A7	4	A12 vs. V1	V1	7
A3 vs. A12	A3	9	A7 vs. V1	V1	7	A12 vs. V2	A12	4
A3 vs. V1	A3	9	A7 vs. V2	A7	5	A12 vs. V3	A12	4
A3 vs. V2	A3	9	A7 vs. V3	A7	5	A12 vs. V4	A12	4
A3 vs. V3	A3	9	A7 vs. V4	A7	5	V1 vs. V2	V1	7
A3 vs. V4	A3	9	A8 vs. A9	A8	4	V1 vs. V3	V1	7
A5 vs. A6	A6	4	A8 vs. A10	A10	7	V1 vs. V4	V1	7
A5 vs. A7	A7	7	A8 vs. A11	A11	4	V2 vs. V3	V2	1
A5 vs. A8	A8	7	A8 vs. A12	A8	3	V2 vs. V4	V2	4
A5 vs. A9	A9	5	A8 vs. V1	V1	5	V3 vs. V4	V3	4
A5 vs. A10	A10	5	A8 vs. V2	A8	5			
A5 vs. A11	A11	6	A8 vs. V3	A8	5			

Tabelle 3.1: Übersicht der erteilten Bewertungen

Platz	Anford.	Wichtigk.	Platz	Anford.	Wichtigk.	Platz	Anford.	Wichtigk.
1	A3	28,59%	6	A8	4,99%	11	V2	1,55%
2	A2	14,20%	7	A9	4,93%	12	V3	1,53%
3	V1	13,04%	8	A6	4,88%	13	A5	1,33%
4	A11	9,30%	9	A7	3,42%	14	V4	1,12%
5	A10	9,15%	10	A12	1,98%	-	-	-

Tabelle 3.2: Übersicht der Prioritäten in Prozent, sortiert nach ihrer Relevanz für den VGBS

3.3.4 Gewichtung und Lösung

In diesem Schritt wird mit Hilfe der im vorhergehenden erfolgten Bewertung ein Index errechnet. Dieser beschreibt die Relevanz der einzelnen Anforderungen für den VGBS in Prozent. In [Tabelle 3.2](#) ist dieser Index nach der Relevanz für den VGBS sortiert dargestellt. Die Prioritäten sind hierbei in Prozent angegeben.

Platz 1: Verlustfreiheit

Beim betrachten von [Tabelle 3.2](#) zeigt sich, dass das verlustfreie Wiederherstellen archivierter Daten mit 28,59% die wichtigste Anforderung ist, welche an das digitale Archiv des VGBS gestellt wird. Der Hintergrund dessen ist, dass ein Archiv, das seine Daten nicht verlustfrei bereitstellen kann keinen Sinn und Zweck erfüllt. Ein Archiv ohne diese Fähigkeit wäre ein Spiel mit dem Zufall bei dem der Nutzer, gerade nach der Aufbewahrungsdauer mehrerer Jahre, rätseln müsste, ob das Informationsobjekt auf das er zugegriffen hat, noch exakt das selbe ist wie vor seiner Archivierung. Es wäre gravierend, wenn nach der Wiederherstellung eines Informationsobjekts ein Informationsverlust zum ursprünglichen Objekt, beispielsweise in Form von fehlenden Nachkommastellen bei einem Beitrag, stattgefunden hätte. Dies würde die Authentizität der archivierten Daten in Frage stellen und wäre insbesondere bei einer anstehenden Außenprüfung fatal.

Platz 2, 3: Direkter Zugriff, Jährlicher Schnappschuss

Weiterhin folgen auf Platz zwei, mit 14,20% und drei, mit 13,04% der direkte Zugriff auf einzelne Informationsobjekte, sowie einen jährlichen Schnappschuss der Vereinsdaten zu archivieren. Der Grund hierfür ist, dass beide Anforderungen im wesentlichen das Ziel der Archivierung darstellen. Jährlich einen Schnappschuss der Vereinsdaten zu archivieren ist essentiell, um im Rahmen einer steuerrechtlichen Außenprüfung oder eines Rechtsstreits die Vereinsdaten eines konkreten Jahres bereitstellen zu können. Damit können unter Umständen schwerwiegende rechtliche Konsequenzen vom VGBS abgewendet werden. Dies ist jedoch nur möglich, wenn der Gesetzgeber einen direkten Zugriff auf die einzelnen Tupel besitzt, die in diesem Schnappschuss gespeichert sind. Der direkte Zugriff ist etwas wichtiger, weil egal welche Daten der Verein, im Kontext seiner Verwaltungsanwendung archiviert, der VGBS sowie der Gesetzgeber Zugriff auf einzelne darin gespeicherte Tupel benötigt.

Platz 4, 5: Unveränderbarkeit, Zugriffsschutz

Die nachfolgenden zwei Plätze werden von der rechtlich geforderten Unveränderbarkeit archivierter Daten mit 9,3% und vom Zugriffsschutz mit 9,15% belegt. Beides

sind Anforderungen, die an die Sicherheit archivierter Daten gestellt werden und somit nur relevant, wenn die oben erwähnten Anforderungen erfüllt sind. Hierbei sind die Unveränderbarkeit sowie der Zugriffsschutz besonders wichtig, weil durch sie gegebenenfalls verhindert werden kann, dass der VGBS bzw. dessen Kunden unmittelbaren Schaden nehmen. Dies könnte beispielsweise der Fall sein, wenn die sensiblen Daten des Vereins (siehe [Abschnitt 3.1.1](#) „Das angestammte Verwaltungssystem“) in die Hände unberechtigter Fallen oder Daten nicht nachweisbar manipuliert werden können.

Platz 6, 7, 8, 9: Agilität, Metadaten, Indexierung, Unabhängigkeit

Das Mittelfeld wird gebildet durch die Agilität mit 4,88%, Metadaten mit 4,37%, die Forderung nach einem eindeutigen Index mit 4,37% und die Unabhängigkeit der archivierten Daten mit 3,45%. Hierbei richten sich die Forderungen nach Agilität und Unabhängigkeit insbesondere an die Nachhaltigkeit des Archivs.

Wie bereits beschrieben kann das Fehlen der Unveränderbarkeit und des Zugriffsschutzes unmittelbare Folgen für den Verein, als auch für seine Kunden haben. Dagegen hätte ein Verlust archivierter Daten, auf Grund eines nicht Nachhaltigen Archivs, nur unmittelbare Folgen für den Verein. Aus diesem Grund sind die Agilität und Unabhängigkeit niedriger priorisiert. Außerdem sind diese im wesentlichen nach ihrer Reihenfolge Priorisiert, mit welcher Wahrscheinlichkeit sie erforderlich werden könnten.

Weiterhin besitzen auch die Indexierung, sowie Metadaten eine geringere Priorität als Platz 2 und 3 dieses Index. Der Hintergrund dessen ist, dass die Folgen wenn die archivierten Daten nicht mit einem Index bzw. Metadaten versehen sind, lediglich diese sind, dass sich die archivierten Daten nur schwer identifizieren bzw. verwalten lassen. Dies ist jedoch nichts verglichen mit den bereits oben beschriebenen Folgen, wenn die archivierten Daten nicht Unveränderbar sind bzw. über keinen Zugriffsschutz verfügen.

Platz 10, 11, 12, 13, 14: Angemessenheit, Verfassen, Bearbeiten, Ordnungskriterien, Löschen

Auf den letzten Plätzen folgen die Forderungen nach Angemessenheit, dem Verfassen sowie Bearbeiten archivierter Datensätze, Ordnungskriterien und dem Löschen archivierter Daten mit 1,98%, 1,55%, 1,53%, 1,33%, 1,12%. Angemessenheit führt diese Liste an, weil sie gesetzlich vorgeschrieben ist. Die Angemessenheit ist jedoch vom Gesetzgeber sehr unpräzise formuliert. Außerdem wird der Verein auch etwas länger auf den Zugriff auf seine archivierten Daten warten können, wenn das Archiv dafür die höher platzierten Anforderungen erfüllt.

Ferner sind Ordnungskriterien nur relevant, um die Übersicht über die archivierten Datensätze zu behalten. Dies ist jedoch nicht so wichtig verglichen mit dem bereits erörterten Nutzen der höher priorisierten Anforderungen wie beispielsweise der Agilität, die dabei hilft Datenverluste zu vermeiden.

Überdies erhielten die Forderungen Daten in archivierten Datenbeständen verfassen, bearbeiten und löschen zu können sehr niedrige Prioritäten, auch wenn sie

direkt vom VGBS gestellt wurden. Der Hintergrund dessen ist, dass sie verglichen zu anderen Anforderungen nicht so wichtig sind. Viele davon sind essentiell wie z.B. die archivierten Daten mit einem Zugriffsschutz zu sichern oder das verlustfreie Wiederherstellen dieser. Deren Vorteile wurden bereits oben besprochen. Andere Anforderungen sind wiederum gesetzlich vorgeschrieben. So zum Beispiel die Unveränderbarkeit. Da der Verein in erster Linie auf Verlangen des Gesetzgebers hin archiviert, sind diese auch höher zu priorisieren, da sonst rechtlichen Strafen folgen könnten. Grundsätzlich gilt jedoch, dass das Verändern archivierter Daten über die Jahre eine organisatorische Notwendigkeit für den Verein geworden ist und deswegen dennoch zwingend umgesetzt werden muss.

3.4 Zusammenfassung

In diesem Kapitel habe ich die verschiedenen Anforderungen, die an das digitale Archiv des VGBS gestellt werden, analysiert. Darunter befinden sich allgemeine Anforderungen, die von gesetzlicher Seite und verschiedenen Autoren gestellt werden, sowie jene die der VGBS selbst an sein digitales Archiv stellt. Im Zusammenhang damit haben sich insbesondere zwei widersprüchliche Anforderungen herausgestellt: Zum einen die gesetzlich geforderte Unveränderbarkeit archivierter Daten. Diese besagt, dass der Ursprungszustand sowie sämtliche Modifikationen archivierter Daten nachvollziehbar sein müssen. Auf der Anderen Seite existiert wiederum die Forderung archivierte Daten verändern zu können, welche direkt vom Verein gestellt wird.

Diese zwei scheinbar gegensätzlichen Anforderungen haben zur Folge, dass nur eine Archivierungslösung in Frage kommt, die Modifikationen an archivierten Daten zulässt, diese jedoch protokolliert. Damit kann der Widerspruch aus der gesetzlich geforderten Unveränderbarkeit und der vom Verein verlangten Veränderbarkeit der archivierten Daten gelöst werden.

Ferner wurden über die Zeit verschiedene Lösungen entwickelt, welche für die Archivierung des VGBS potentiell in Frage kommen. In diesem Zusammenhang wären zunächst das SIARD-Format, sowie die Database Markup Language genannt. Beide Ansätze basieren darauf die Struktur und die Daten einer Datenbank mit Hilfe von XML-Derivaten zu archivieren [JLRH02, Inn08]. In diesem Kontext ist es wiederum auch möglich einen Dump der Datenbank durchzuführen. Ein weiterer denkbarer Ansatz bestünde darin das System der bisherigen Archivierung des Vereins weiterzuführen. Das heißt bei Jahreswechsel die operative Datenbank zu duplizieren und eine Version zu archivieren. Die andere Version wird weiterhin operativ genutzt. Außerdem möglich wären Formen der temporalen Datenhaltung zu nutzen.

Betrachtet man diese Lösungen, so existieren im wesentlichen zwei Ansätze: Einerseits den datenbankinternen Ansatz, bei dem die archivierten Daten innerhalb des DBS aufbewahrt werden. Andererseits den datenbankexternen Ansatz, im Kontext dessen diese in Dateien außerhalb des DBS gespeichert werden.

Keine der oben besprochenen Lösungen, die diesen Ansätzen entsprechen, bietet von sich aus die bereits ausgeführte Fähigkeit Daten zu archivieren und an diesen Daten durchgeführte Modifikationen zu protokollieren. Aus diesem Grund wird es erforderlich, auf diesen Ansätzen aufbauend, eigene Methoden zu konzipieren. Deswegen werde ich im folgenden Kapitel 4 die datenbankinterne und datenbankexterne Methode vorstellen.

4. Vorstellung der Archivierungsmethoden

Im vorhergehenden Kapitel 3 hat sich gezeigt, dass zwei widersprüchliche Anforderungen an das digitale Archiv des VGBS gestellt werden: Zum einen die Anforderungen des VGBS seine archivierten Vereinsdaten verändern zu können (siehe [Abschnitt 3.2.2 „VGBS-Anforderungen“](#)). Auf der anderen Seite die vom Gesetzgeber geforderte Unveränderbarkeit der archivierten Daten (siehe [Abschnitt 3.2.1.11 „Unveränderbarkeit“](#)).

Als Konsequenz daraus werde ich in diesem Kapitel die datenbankinterne und datenbankexterne Methode vorstellen. Beide Archivierungsmethoden wurden insbesondere so konzipiert, dass sie den oben angedeuteten Widerspruch aus Veränderbarkeit und Unveränderbarkeit aus konzeptueller Sicht lösen können.

Im Rahmen der datenbankinternen Methode befinden sich die archivierten Daten innerhalb der Datenbank, wobei das Archivieren mit Hilfe des Datenbankschemas und der Anwendung erfolgt. Im Gegensatz dazu findet die Archivierung im Kontext der datenbankexternen Methode mit Hilfe von inkrementellen Backups in Form von protokollierten SQL-Anweisungen statt. Diese werden in Dateien an einem Speicherort außerhalb des DBS gelagert.

Für die Vorstellung beider Methoden werde ich zunächst ein begleitendes Beispiel einführen. Dieses werde ich nutzen, um die Funktionsweise, sowie die Fähigkeiten der datenbankexternen und - internen Methode in Hinblick auf die eingangs erwähnten Anforderungen des Vereins und die rechtlich vorgeschriebene Unveränderbarkeit zu erläutern.

Im Anschluss daran werde ich dann beide Methoden unter zu Hilfenahme dieses Beispiels vorstellen. Dazu werde ich zunächst deren allgemeine Eigenschaften beschreiben. Im Nachfolgenden werde ich die für die datenbankinterne und -externe Methode essentiellen Konzepte einführen und deren Funktionsweise am zuvor eingeführten Beispiel erläutern. Dabei werde ich insbesondere ausführen wie der oben

besprochene Widerspruch aus Unveränderbarkeit und Veränderbarkeit von beiden Methoden gelöst wird.

4.1 Begleitendes Beispiel

Das nachfolgende Beispiel wird mich bei der Vorstellung der datenbankinternen und -externen Methode begleiten. Dabei dient es in erster Linie dem Zweck die Möglichkeiten und Fähigkeiten beider Methoden, in Bezug auf die Anforderungen des VGBS nach Veränderbarkeit, sowie der rechtlichen Forderung nach Unveränderbarkeit des archivierten Datenbestandes, darzustellen.

In diesem Beispiel wird eine Situation dargestellt, in der diese Anforderungen von zentraler Bedeutung sind. Das Beispiel wird anhand einer zeitlichen Abfolge von Geschehnissen beschrieben. Diese Geschehnisse sind zum einen für die Funktionalität des Archivs relevant. Auf der anderen Seite bilden sie die zeitlichen Eckdaten für die Protokollierung der erfolgten Datenmodifikationen.

Für die Illustration dieses Szenarios verwende ich die Assoziationen zwischen den Tabellen „Mitglied“ und „Sportgruppe“, sowie „Mitglied“ und „Beitrag“ (siehe [Abbildung 4.1](#)). Auf Grund der Tatsache, dass diese Tabellen über zu viele Attribute zur Darstellung in diesem Kontext verfügen, welche außerdem zur Illustration dieses Beispiels nicht relevant sind, werden die hier aufgeführten Tabellen reduziert. Das vollständige findet sich in [Abbildung A.7](#) in Anhang A.

Dies bedeutet, dass „Mitglied“ nur noch den Primärschlüssel „MitglNr“, das Nichtschlüsselattribut „Name“ und den Fremdschlüssel „Beitrag“ umfasst. Ähnlich sieht es bei der Tabelle „Sportgruppe“ aus: Diese wird auf den Primärschlüssel „SgNr“ und ebenfalls auf das Nichtschlüsselattribut „Name“ reduziert. Weiterhin handelt es sich bei der Assoziation dieser beiden Tabellen um eine N:M-Beziehung, weshalb der Beziehungstyp „Ist in“ auch eine Tabelle beschreibt. Diese umfasst den Primärschlüssel „MSgNr“, die zwei Fremdschlüssel „MitglNr“ und „SgNr“, sowie die Attribute „Beitritt“ und „Austritt“. Diese zwei letzten Attribute bilden den Beitritts- und Austrittszeitpunkt eines Mitglieds in und aus einer Sportgruppe ab. Dabei ist „Beitritt“ ein weiterer Teil des Primärschlüssels. Des Weiteren setzt sich die Tabelle „Beitrag“ aus dem Primärschlüssel „Name“ und den Nichtschlüsselattributen „Entspricht“ und „Betrag“ zusammen. In diesem Zusammenhang beschreibt das Attribut „Entspricht“, in welcher Sportgruppe ein Mitglied eingeschrieben sein muss, damit er diesen Beitrag zugewiesen bekommt. Über dies besteht zwischen „Mitglied“ und „Beitrag“ eine 1:M-Beziehung.

Beispiel

Danny ist neu im VGBS und wird am 01.07.15 der Sportgruppe „Sanfte Ganzkörperfitness“ (SGKF) beitreten. Der Beitrag den Danny nun zahlen muss errechnet sich aus mehreren Faktoren, weshalb ihm dieser separat zugeteilt wird. Um es für dieses Beispiel unkompliziert zu halten, werden die möglichen Beiträge mit A und B bezeichnet. Unter den Beitrag A fallen Mitglieder, welche die Sportgruppe „SGKF“ gewählt haben. Dahingegen müssen Mitglieder, die sich für die Sportgruppe „Freizeitsport“ entschieden haben Beitrag B zahlen.

Mitglied			Ist in					Sportgruppe	
MitglNr	Name	Beitrag	MSgNr	MitglNr	SgNr	Beitritt	Austritt	SgNr	Name
1	Danny	B	1	1	1	01.07.15	-	1	SGKF
								2	Freizeitsport
								3	FitAb50+

Beitrag		
Name	Entspricht	Betrag
A	SGKE, FitAb50+	20€
B	Freizeitsport	50€

Abbildung 4.1: 25.05.15 - 12:49:34 Uhr: Tim schreibt Danny in die Sportgruppe „SGKF“ ein.

25.05.15 um 12:49:34 Uhr: Der VGBS-Mitarbeiter Tim trägt Danny, zusammen mit seiner Zuweisung zur Sportgruppe „SGKF“, in die Datenbank des VGBS ein. Dabei teilt er Danny versehentlich den Beitrag B zu. Damit benachteiligt er Danny, weil dieser somit monatlich 30 Euro mehr zahlen muss. Die ist in [Abbildung 4.1](#) illustriert.

07.01.15: Danny tritt der Sportgruppe „SGKF“ bei und beginnt monatlich die 50 Euro der Sportgruppe „Freizeitsport“ zu zahlen.

01.01.16: Der VGBS will den jährlichen Schnappschuss seiner Vereinsdaten archivieren. Deswegen werden zum Jahreswechsel Dannys falsche Beitragszuweisung sowie gezahlte Beiträge, zusammen mit den anderen Daten des VGBS, in das Archiv aufgenommen.

08.04.16: Zum 01.05.16 wird Danny aus Altersgründen aus der Sportgruppe „SGKF“ austreten. Der VGBS möchte diesen Fakt daher auch in seiner Datenbank festhalten. Im Rahmen dessen stellt Tim fest, dass er Danny für den Zeitraum vom 07.01.15 bis zum 01.05.16, in dem dieser in der Sportgruppe „SGKF“ war, den Beitrag B zugewiesen hatte. Dieser entspricht der Sportgruppe „Freizeitsport“, wobei Danny über diesen Zeitraum eigentlich Beitrag A hätte zahlen müssen.

Statt eine Entlassung zu riskieren und um dem Verein die Rückzahlung der erhöhten Beiträge oder einen Rechtsstreit zu ersparen, entscheidet sich Tim dieses Problem eigenständig zu korrigieren.

08.04.16 um 12:45:56 Uhr: Wie in [Abbildung 4.2](#) ersichtlich notiert Tim dafür zunächst in der Datenbank, dass Danny am 01.05.16 austreten wird und schreibt ihm parallel dazu für den Zeitraum vom 01.07.15 bis 01.05.16 in die Sportgruppe „Freizeitsport“ ein. Damit sieht es im operativen Datenbestand des Vereins so aus, als wäre Danny niemals in der Sportgruppe „SGKF“ gewesen.

08.04.16 um 12:47:12 Uhr: Tim weiss, dass der VGBS zum Jahreswechsel seine Daten archiviert. Deswegen schreibt er Danny auch in den archivierten Daten in die Sportgruppe „Freizeitsport“ ein.

Wenn der Verein nun die Möglichkeit hätte diese nachträgliche Veränderung nachzuvollziehen dann könnte er Tim entlassen und die Rückzahlung der zu hoch gezahlten

Mitglied			Ist in					Sportgruppe	
MitglNr	Name	Beitrag	MSgNr	MitglNr	SgNr	Beitritt	Austritt	SgNr	Name
1	Danny	A	1	1	2	01.07.15	01.05.16	1	SGKF
								2	Freizeisport
								3	FitAb50+

Beitrag		
Name	Entspricht	Betrag
A	SGKF, FitAb50+	20€
B	Freizeitsport	50€

Abbildung 4.2: 08.04.16 - 12:45:56 Uhr: Tim vertuscht seinen Fehler im operativen Datenbestand.

Beiträge einleiten. Somit wären die gegensätzlichen Anforderungen vereinbar. Im Folgenden werden die interne und externe Methode vorgestellt und genau auf diesen Sachverhalt hin konzipiert bzw. untersucht.

4.2 Datenbankinterne Methode

In *Abbildung 4.3* kann der Aufbau der datenbankinternen Methode betrachtet werden. Diese zeichnet sich dadurch aus, dass die archivierten Daten innerhalb der operativen Datenbank aufbewahrt und verwaltet werden. Sie befinden sich also noch unter der Kontrolle des DBMS. Weiterhin findet die Archivierung der Daten mit Hilfe des Datenbankschemas und der Anwendung statt.

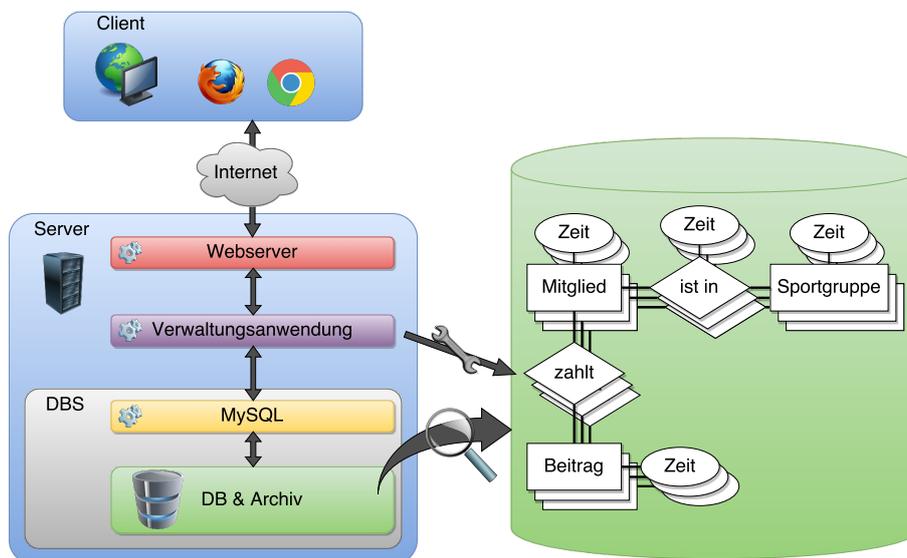


Abbildung 4.3: Datenbankinterne Methode: Übersicht

Das Datenbankschema wird dabei um zeitliche Attribute ergänzt, mit denen die Evolution, sowie die Korrekturhistorie eines Datensatzes abgebildet werden kann. Mit Hilfe der Daten-Evolution ist es möglich, die für den VGBS relevanten Daten zu archivieren und modifizieren. Ergänzend dazu ermöglicht die Korrekturhistorie,

diese Modifikationen nachzuvollziehen und im Kontext einer steuerrechtlichen Außenprüfung oder eines Rechtsstreits auswerten zu können. Somit wird mit dessen Hilfe die Unveränderbarkeit der archivierten Daten erreicht.

Ferner wird im Falle einer Modifikation eines Datensatzes dieser nicht mehr überschrieben oder gelöscht, sondern seine modifizierte Version der Tabelle hinzugefügt. Die ursprüngliche unveränderte Version bleibt dabei in der Tabelle erhalten. Dadurch entsteht, neben den Tupeln und Attributen, eine dritte Dimension: die Zeit.

Diese erweiterte Logik, sowie die Einhaltung daraus resultierender zusätzlicher Integritätsbedingungen, wird durch die Anwendung umgesetzt. Dies geschieht mit der Unterstützung von Funktionen, die vom DBMS bereitgestellt werden. Dabei erfolgt die Trennung der zeitlichen Daten ebenfalls durch die Anwendung mittels SQL-Operationen in Form von eingeschränkten Selektionsprädikaten.

4.2.1 Konzepte

Zur Umsetzung der datenbankinternen Methode werde ich zum einen das Konzept des Verwaltungsjahres nutzen. Auf der anderen Seite kommen Konzepte der temporalen Datenhaltung zum Einsatz (siehe [Abschnitt 2.2](#) „Temporale Datenhaltung“). Insbesondere sind dies die Gültigkeitszeit sowie Transaktionszeit. Diese werden hierfür zu einer bitemporalen Relation kombiniert. Die Gültigkeitszeit wurde dabei extra für die Anforderungen des VGBS angepasst und wird somit im folgenden als VGBS-Gültigkeitszeit bezeichnet.

4.2.1.1 Verwaltungsjahr

Ein jährlicher Schnappschuss der Vereinsdaten wird im Folgendem stets als **Verwaltungsjahr** bezeichnet. Damit wird kenntlich, dass es sich hierbei um den Datenbestand eines Jahres handelt, der für die Verwaltung des VGBS für ein bestimmtes Jahr relevant und damit operativ ist bzw. war. Ferner handelt es sich beim **operativen Verwaltungsjahr**, um jenen Datenbestand der zurzeit operativ genutzt wird.

4.2.1.2 VGBS-Gültigkeitszeit

Die VGBS-Gültigkeitszeit gibt dem Verein die Möglichkeit den letzten Schnappschuss seiner Daten zu erhalten und erlaubt ihm historische Daten nachträglich zu verfassen, bearbeiten und zu löschen. Somit werden mit Hilfe der VGBS-Gültigkeitszeit die im [Abschnitt 3.2.2](#) erläuterten Anforderungen des Vereins realisiert.

Die Gültigkeitszeit beschreibt die Zeit, wann ein Fakt in der realen Welt als wahr bzw. gültig angenommen wird [[JDB⁺98](#)]. Mit der Gültigkeitszeit kann die Evolution von Fakten dargestellt werden. Hierbei wird diese stets vom Nutzer angegeben und kann somit vergangene, gegenwärtige oder zukünftige Zeiträume umfassen. Näheres zur Gültigkeitszeit wurde in [Abschnitt 2.2](#) „Gültigkeitszeit“ erläutert.

Im Kontext der Verwaltungsanwendung des VGBS wird die Gültigkeitszeit dazu verwendet, auszudrücken über welchen Zeitraum ein in der Datenbank eingetragener Fakt für den VGBS organisatorisch relevant und damit operativ ist bzw. war. Damit können nicht mehr operativ relevante Daten mit Hilfe der Verwaltungsanwendung

logisch von den operativen getrennt und somit archiviert werden. Auf Grund dieser Anpassung der Gültigkeitszeit an die Bedürfnisse des VGBS wird im Rahmen dieser Arbeit von der VGBS-Gültigkeitszeit gesprochen.

Die VGBS-Gültigkeitszeit wird durch ein rechtsseitig halboffenes zeitliches Intervall abgebildet. Die zusätzlichen Integritätsbedingungen die sich im Zusammenhang mit der normalen Gültigkeitszeit aus dieser Abbildungsform ergeben, gelten uneingeschränkt auch für die VGBS-Gültigkeitszeit (siehe [Abschnitt 2.2.2.3](#) „Integritätsbedingungen“). Ferner wird das zeitliche Intervall der VGBS-Gültigkeitszeit durch die Attribute „GültigSeit“ und „GültigBis“ repräsentiert. „GültigSeit“ charakterisiert den Beginn und „GültigBis“ das Ende dieses Intervalls. Ersteres wird dabei mit in den Primärschlüssel aufgenommen.

Hierbei zeigt das Attribut „GültigSeit“ immer auf das Verwaltungsjahr zu dem ein Datensatz ursprünglich für den VGBS operativ von Nutzen war. Dagegen beschreibt das Attribut „GültigBis“ bis zu welchem Verwaltungsjahr ein Datensatz operativ war bzw. sein wird. Dementsprechend liegt dieses Intervall in der Granularität von Jahren vor. Ein Datensatz ist operativ, wenn sein Gültigkeitsende auf einen unbegrenzt entfernten Zeitpunkt festgelegt wurde. Dieser wird durch das Jahr 2143 dargestellt, welcher der maximal mögliche Wert des Datentyps „YEAR“ in MySQL ist. Im Gegensatz dazu wird ein Datensatz als archiviert betrachtet, wenn das Ende seiner VGBS-Gültigkeitszeit ein bestimmtes Verwaltungsjahr beschreibt.

4.2.1.3 Transaktionszeit

Die Transaktionszeit beschreibt die Zeit zu der ein Datensatz in die Datenbank eingefügt und logisch gelöscht wurde [[JDB+98](#)]. Näheres zur Transaktionszeit findet sich in [Abschnitt 2.2.1.10](#). Die Transaktionszeit bildet die Korrekturhistorie der Daten ab und gewährleistet damit die gesetzlich vorgeschriebene Unveränderbarkeit archivierter Daten.

Im Kontext der datenbankinternen Methode wird die Transaktionszeit durch ein rechtsseitig halboffenes Intervall beschrieben. Dies liegt in der Granularität von Sekunden vor. Damit kann jede Modifikation von Daten sekunden-genau protokolliert werden. Dieses Intervall wird durch die beiden Attribute „ErfasstAm“ und „GelöschtAm“ ausgedrückt. „ErfasstAm“ zeigt auf den Beginn dieses Intervalls. Dahingegen beschreibt „GelöschtAm“ dessen Ende und wird mit in den Primärschlüssel der Relation aufgenommen. Ein noch unbegrenztes Intervallende wird durch den Wert „9999-12-31 23:59:59“ symbolisiert. Dies ist der maximal mögliche Wert für den Datentyp DATETIME in MySQL.

4.2.2 Datentrennung

Die Trennung von operativen und archivierten Daten bzw. aktuellen und jenen die dies nicht mehr sind erfolgt, wie bereits eingangs besprochen, durch eine SQL-Anweisung in Form eines eingeschränkten Selektionsprädikats. Diese sieht wie folgt aus:

```
SELECT * FROM Tabelle
WHERE Gültigkeitsbeginn <= Verwaltungsjahr
AND Gültigkeitsende > Verwaltungsjahr ;
AND ErfasstAm <= Transaktionszeitpunkt
AND GelöschtAm > Transaktionszeitpunkt
```

Wie ersichtlich wird dieses Selektionsprädikat hinsichtlich eines Verwaltungsjahres und Transaktionszeitpunktes eingeschränkt. Hierbei spezifiziert der Nutzer in der Anwendung zunächst das Verwaltungsjahr, das er betrachten möchte. Dabei werden ihm nur jene Datensätze bereitgestellt, deren Gültigkeitsbeginn kleiner oder gleich und Gültigkeitsende größer als das spezifizierte Verwaltungsjahr ist. Da operative Datensätze ein unbegrenztes Gültigkeitsende besitzen, ist dieses stets größer als das spezifizierte Verwaltungsjahr. Dadurch stehen diese in jedem Verwaltungsjahr zur Verfügung, solange ihr Gültigkeitsbeginn kleiner gleich dem Verwaltungsjahr ist. Im Gegensatz dazu besitzen archivierte Daten ein in der Vergangenheit liegendes Gültigkeitsende und werden im operativen Datenbestand nicht angezeigt. Der Grund hierfür ist, dass deren Gültigkeitsende immer kleiner als das operative Verwaltungsjahr ist.

Im Rahmen dessen werden dem Nutzer in Regel nur aktuelle Datensätze angezeigt. Häufig sind nur diese für den Nutzer von Interesse. Möchte der Nutzer zusätzlich wissen wie ein in einem bestimmten Verwaltung gültiger Datensatz zu einem bestimmten Zeitpunkt beschaffen war, so spezifiziert der Nutzer einen Transaktionszeitpunkt. Daraufhin wird ihm die Version des Datensatzes bereitgestellt die im jeweiligen Verwaltungsjahr zum spezifizierten Transaktionszeitpunkt aktuell war. Also jene dessen Transaktionsbeginn kleiner gleich und Transaktionsende größer als der Transaktionszeitpunkt ist.

4.2.3 Beispiel

Im Folgendem werde ich die Fähigkeiten der datenbankinternen Methode hinsichtlich den Anforderungen des VGBS sowie der gesetzlich vorgeschriebenen Unveränderbarkeit erläutern. Außerdem werde ich einen Einblick in die Funktionsweise der datenbankinternen Methode geben. Detailliert kann diese im Anhang A in [Abschnitt A.2.1](#) nachgeschlagen werden. Zu diesem Zweck werde ich das in [Abschnitt 4.1](#) hierfür eingeführte Beispiel heranziehen.

Der Gültigkeitsbeginn eines neu hinzugefügten Datensatzes wird stets auf das Verwaltungsjahr festgelegt, in dem dieser hinzugefügt wurde. Dadurch wird es möglich Daten in einem archivierten Datenbestand zu verfassen, welches eine Anforderung des VGBS ist. Weiterhin erhalten neu hinzugefügte Datensätze ein unbestimmtes Gültigkeitsende und sind damit operativ. Außerdem wird diesen ein aktueller Transaktionszeitstempel verpasst. Dieser zeigt exakt auf den realen Zeitpunkt zu dem ein



Abbildung 4.4: Tim fügt Danny der Datenbank hinzu.

Datensatz der Datenbank hinzugefügt wurde. Damit wird erkenntlich, dass es sich hierbei um die erste und ursprüngliche Version eines Datensatzes handelt, so wie es von der Unveränderbarkeit gefordert ist.

In der Regel ist die Verwaltungsanwendung des VGBS auf das operative Verwaltungsjahr eingestellt. So war es auch, als Tim am **25.05.15 um 12:49:34 Uhr** Danny der Datenbank des Vereins hinzugefügt hat. Dementsprechend war dabei das Verwaltungsjahr 2015. Wie in [Abbildung 4.4](#) illustriert erhalten somit der Datensatz Danny, zusammen mit seiner Zuweisung zur Sportgruppe „SGKF“ den Gültigkeitszeitstempel [2015, 2143]. Der Transaktionszeitstempel wurde hierbei auf [25.05.15 12:49:34, 9999-12-31 23:59:59] festgelegt.

Wird ein Datensatz in einem Verwaltungsjahr bearbeitet, das ungleich seines Gültigkeitsbeginn ist, so wird eine neue bearbeitete Version des gleichen Datensatzes der Tabelle hinzugefügt. Diese ist fortan von diesem Verwaltungsjahr an gültig. Die Gültigkeit der ursprünglichen unveränderten Version endet dahingegen in diesem Verwaltungsjahr. Damit bleiben die Daten der vorhergehenden Verwaltungsjahre erhalten, so wie der Verein dies fordert. Durch die Natur der Transaktionszeit führt in diesem Zusammenhang jede Veränderung von Gültigkeitszeiten, ebenfalls zu einer weiteren von dort an aktuellen Version des Datensatzes. Damit wird sichergestellt, dass jegliche Veränderung von Daten protokolliert wird, so wie es die Unveränderbarkeit fordert.

Die folgenden Ausführungen zu Tims Manipulationen bezüglich Dannys Sportgruppenzuweisung können in [Abbildung 4.5](#) betrachtet werden. Veränderungen der Daten werden in grau hinterlegt dargestellt. Jedes Tupel erhält einen Buchstaben von A bis D. A beschreibt dabei Dannys ursprüngliche Zuweisung zur Sportgruppe „SGKF“.

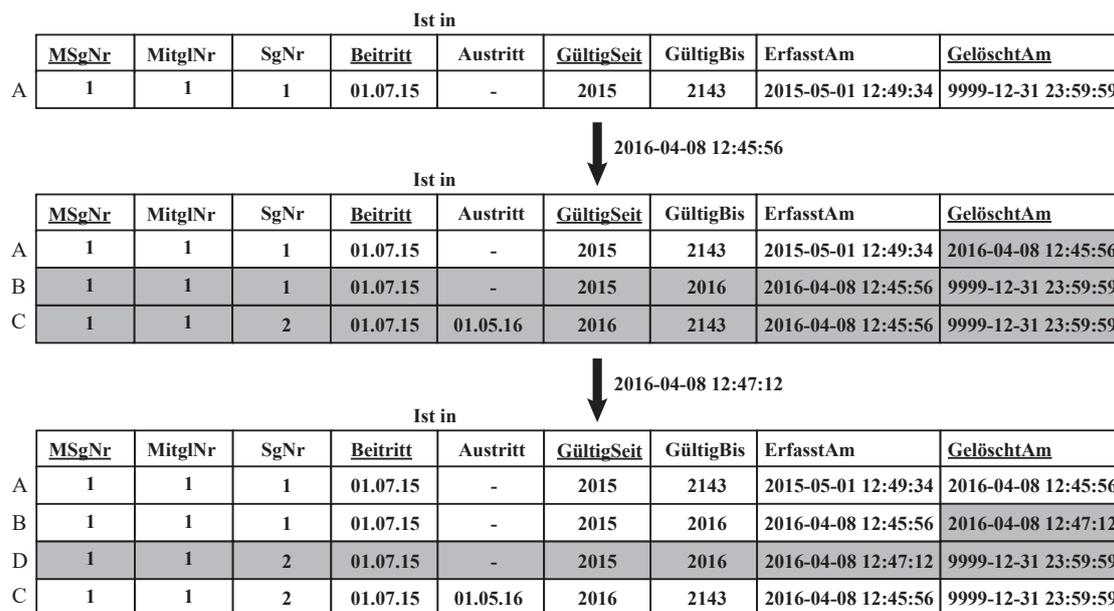


Abbildung 4.5: Ablauf der Geschehnisse

Zum 01.01.16 wechselt das operative Verwaltungsjahr von 2015 zu 2016. Am **08.04.16 um 12:45:56 Uhr** vermerkt Tim, dass Danny am 01.05.16 aus der Sportgruppe „SGKF“ austreten wird. Parallel dazu trägt er Danny für den Zeitraum vom 01.07.15 bis 01.05.16 in die Sportgruppe „Freizeitsport“ ein. Dies bedeutet, dass der mit dem Buchstaben A gekennzeichnete Datensatz im Verwaltungsjahr 2016 bearbeitet wird.

Da der Gültigkeitsbeginn von A 2015 ist, wird eine neue Version in die Tabelle eingefügt, welche ab dem Verwaltungsjahr 2016 gültig ist. Dafür wird zunächst das Gültigkeitsende von Datensatz A auf 2016 festgelegt. Weil es sich hierbei um eine Modifikationen handelt, bewirkt die Transaktionszeit, dass ein neuer Datensatz der Tabelle hinzugefügt wird. In diesem Fall ist dies Datensatz B. Dieser beinhaltet die Modifikation und erhält einen aktuellen Transaktionszeitstempel. Parallel dazu wird das Transaktionsende von A auf den Zeitpunkt der Modifikation festgelegt. Damit ist A nicht mehr aktuell.

Im Anschluss daran wird Datensatz C hinzugefügt, welcher Dannys Zuweisung zur Sportgruppe „Freizeitsport“ für das Verwaltungsjahr 2016 beinhaltet. Diese behält das Gültigkeitsende der ursprünglichen Version A, also 2143 und ist fortan ab dem Verwaltungsjahr 2016 gültig. Damit diese Veränderung protokolliert wird, erhält C ebenfalls einen aktuellen Transaktionszeitstempel.

Ein weiteres Szenario besteht darin, dass der Nutzer einen Datensatz bearbeitet dessen Gültigkeitsbeginn gleich dem spezifizierten Verwaltungsjahr ist. Hierbei wird dieser Datensatz mit einer bearbeiteten neuen Version überschrieben. Die Transaktionszeit bewirkt dabei, dass dieses Überschreiben nur logisch stattfindet. Das heißt die bearbeitete neue Version wird der Tabelle als eigenständiges Tupel mit aktuellem Transaktionszeitstempel hinzugefügt. Der Transaktionsbeginn des neuen Tupels

sowie das Transaktionsende des ursprünglichen zeigen dabei auf den Zeitpunkt der Modifikation.

Am **08.04.16 um 12:47:12 Uhr** manipuliert Tim Dannys archivierte Daten, damit auch diese so aussehen, als wären seine zu hoch gezahlte Beiträge gerechtfertigt. Hierfür wechselt Tim mit Hilfe der Verwaltungsanwendung in das Verwaltungsjahr 2015. Dort wird ihm der in [Abbildung 4.5](#) dargestellte Datensatz B bereitgestellt. Nun bearbeitet Tim diesen und ändert dessen Sportgruppenreferenz auf „Freizeitsport“. Daraus folgt, dass Version D der Tabelle hinzugefügt wird. Diese erhält einen aktuellen Transaktionszeitstempel. Die ursprüngliche Version B ist noch in der Tabelle enthalten und erhält als Transaktionsende den Zeitpunkt der Modifikation 08.04.16 12:47:12. Da sowohl Version B, als auch C den gleichen Gültigkeitszeitstempel aufweisen, überschreibt C die Version B im Verwaltungsjahr 2015.

Hierdurch wird ein Fall beschrieben, wie es mit Hilfe der VGBS-Gültigkeitszeit möglich ist archivierte Daten zu verändern, wie es vom Verein gefordert wird. Außerdem wird hierbei deutlich wie die Transaktionszeit die Unveränderbarkeit der archivierten Daten sicherstellt. Dadurch kann der VGBS durch das Betrachten der einzelnen Versionen des Datensatzes mittels SQL-Anweisungen, letztlich Tims Manipulationen aufspüren.

4.3 Datenbankexterne Methode

Die Übersicht zur datenbankexternen Methode kann in [Abbildung 4.6](#) betrachtet werden. Das besondere Merkmal dieser Archivierungsmethode ist, dass die archivierten Daten in Dateien außerhalb des Datenbanksystems gelagert werden. Hierbei wird ein archivierter Datensatz durch eine Folge von zeitlich indexierten SQL-Anweisungen repräsentiert. Diese entstehen durch das kontinuierliche Protokollieren von Modifikationen in der Datenbank befindlicher Daten. Dies erfüllt im wesentlichen folgende Zwecke:

Zum einen kann mit Hilfe dieser SQL-Anweisungen der Datenbestand eines vorhergehenden Jahres archiviert werden, so wie es der Verein fordert. Hierzu werden diese in den oben erwähnten Dateien gespeichert. Durch die Summe der in ihnen gespeicherten SQL-Anweisungen repräsentiert jede dieser Dateien einen jährlichen Schnappschuss der Vereinsdaten. Hierbei entscheidet der Verein durch die Spezifikation eines Jahres in der Verwaltungsanwendung, welchen dieser Schnappschüsse er betrachten möchte. Danach entscheidet sich auch in welcher Datei die Modifikationen protokolliert werden. Damit werden die Anforderungen des VGBS seinen archivierten Datenbestand im vollem Umfang modifizieren zu können erfüllt (siehe [Abschnitt 3.2.2 „VGBS-Anforderungen“](#)).

Auf der anderen Seite wird mit der durchgängigen Protokollierung von Modifikationen sowie deren zeitliche Indexierung die rechtlich geforderte Unveränderbarkeit (siehe [Abschnitt 3.2.1.11 „Unveränderbarkeit“](#)) sichergestellt.

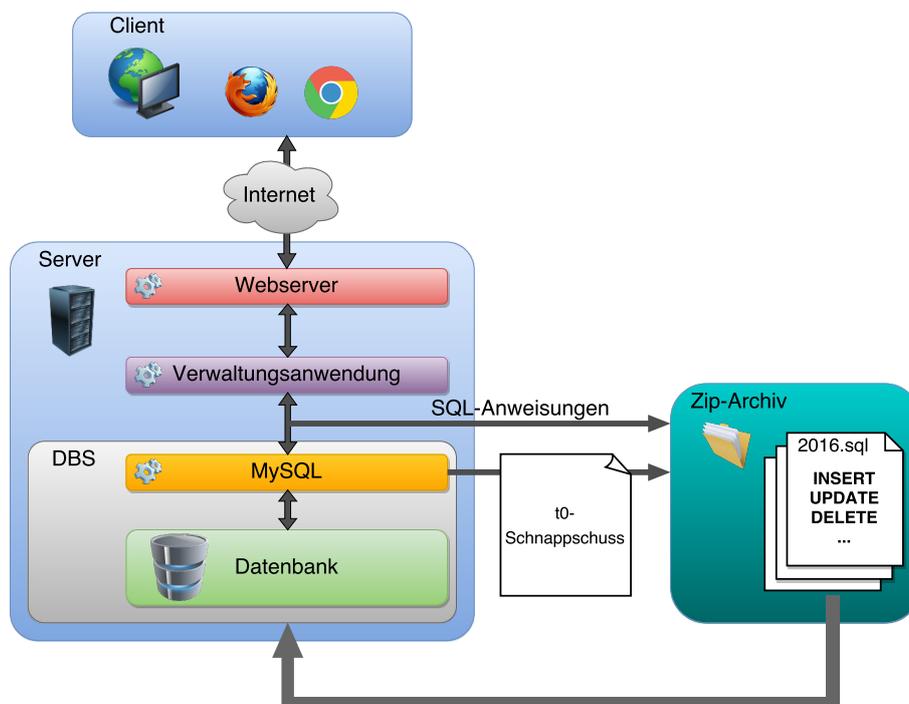


Abbildung 4.6: Datenbankexterne Methode: Übersicht

4.3.1 Konzepte

Bevor ich die datenbankexterne Methode an einem Beispiel illustriere, werde ich in diesem Abschnitt die dafür relevanten Konzepte vorstellen. Diese umfassen das Verwaltungsjahr und die Modifikationszeit.

4.3.1.1 Verwaltungsjahr

Analog zur datenbankinternen Methode wird der jährliche Schnappschuss der Vereinsdaten auch in diesem Kontext als **Verwaltungsjahr** bezeichnet. Hierbei stellt das **operative Verwaltungsjahr** ebenfalls den aktuellen Schnappschuss dar. Im Kontext der datenbankexternen Methode wird ein Verwaltungsjahr durch eine Log-Datei repräsentiert, welche den Namen des dazugehörigen Jahres trägt (siehe [Abbildung 4.6](#)). Hierbei werden alle Log-Dateien in einem passwortgeschützten Zip-Archiv verwaltet. In einer Log-Datei werden untereinander sämtliche SQL-Anweisungen protokolliert, die für das jeweilige Jahr für den Verein organisatorisch relevant sind bzw. waren. Dadurch kann der VGBS gezielt Datenbestände aus vorhergehenden Jahren wiederherstellen und diese modifizieren. Aus diesem Grund werden hiermit sämtliche Anforderungen des VGBS erfüllt.

4.3.1.2 Modifikationszeit

Die Modifikationszeit dient in erster Linie der Sicherstellung der rechtlich geforderten Unveränderbarkeit der archivierten Daten des Vereins. Angelehnt an die Transaktionszeit beschreibt die Modifikationszeit somit den Zeitpunkt, an dem eine Änderung von Datensätzen stattgefunden hat. Damit lässt sich feststellen ob diese ursprünglich oder nachträglich getätigt wurden. Aus diesem Grund wird die Modifikationszeit im Falle einer Datenmodifikation direkt vom System vergeben und ist dadurch an die Gegenwart geknüpft.

Jede SQL-Anweisung wird hierbei mit einem Modifikationszeitstempel versehen, welcher mit der Granularität von Sekunden angegeben ist. Nur diese Granularität bietet eine optimale Nachvollziehbarkeit der Reihenfolge durchgeführter Modifikationen. Dabei zeigt dieser Zeitstempel auf, wann eine SQL-Anweisung von der Verwaltungsanwendung zum DBMS gesendet wurde. Nachfolgend ist die SQL-Anweisung zusammen mit ihrem entsprechenden Modifikationszeitstempel illustriert, mit welcher Danny im Beispiel aus [Abschnitt 4.1](#) in die Datenbank des VGBS aufgenommen wurde.

```
/* Modifikationszeitpunkt: 25.05.2015 - 12:49:34 */  
INSERT INTO Mitglied VALUES ( 'Danny', 'B' );
```

4.3.2 Beispiel

In diesem Abschnitt werde ich verdeutlichen, wie die datenbankexterne Methode den Widerspruch aus den Anforderungen des Vereins und der gesetzlich geforderten Unveränderbarkeit löst. Zudem werde ich einen Überblick der Funktionsweise der datenbankexternen Methode geben. Diese kann detailliert im Anhang A in [Abschnitt A.2.2](#) betrachtet werden. In diesem Kontext werde ich wieder mein in [Abschnitt 4.1](#) eingeführtes Beispiel verwenden.

Der Nutzer spezifiziert in der Verwaltungsanwendung, welches Verwaltungsjahr er betrachten möchte. Dies entscheidet darüber, welche Log-Datei wiederhergestellt wird. Wurde die Log-Datei eines Verwaltungsjahres wiederhergestellt, so betrachtet der Nutzer das ausgewählte Verwaltungsjahr. Jede Modifikation die der Nutzer daran vornimmt, wird nur in der entsprechenden Log-Datei protokolliert. Damit wird die vom Verein geforderte Veränderbarkeit der archivierten Vereinsdaten gewährleistet.

Im Regelfall befindet sich die Verwaltungsanwendung im operativen Verwaltungsjahr. Dies war 2015, als Tim am **25.05.15 um 12:49:34 Uhr** Dannys Datensatz zusammen mit seiner Zuweisung zur Sportgruppe „SGKF“ in die Datenbank des VGBS eingetragen hat. Somit wurde die dazugehörige INSERT-Anweisung in die Datei *2015.sql* eingetragen und ihr der entsprechende Modifikationszeitstempel gegeben. In der Datei *2015.sql* wurde nun folgendes protokolliert:

```
/* Modifikationszeitpunkt: 25.05.2015 - 12:49:34 */
INSERT INTO 'Mitglied' VALUES ('Danny', 'B');

/* Modifikationszeitpunkt: 25.05.2015 - 12:49:34 */
INSERT INTO 'Ist_in' VALUES (1, 1, 1, 01.07.15, '-');
```

Hierbei ist ersichtlich, dass die Modifikationszeit dieser SQL-Anweisungen immer auf den genauen Zeitpunkt des Einfügens der Datensätze in die Datenbank zeigt.

Am 01.01.16 wechselt das operative Verwaltungsjahr von 2015 zu 2016. Das bedeutet, dass nun sämtliche im operativen Verwaltungsjahr durchgeführten Modifikationen in der Log-Datei *2016.sql* protokolliert werden. Dahingegen wird *2015.sql* als archiviert betrachtet.

Dementsprechend wird die UPDATE-Anweisung, mit der Tim am **08.04.16 um 12:45:56 Uhr** Dannys Zuweisung zur Sportgruppe „SGKF“ vertuscht in der Logdatei *2016.sql* vermerkt. In dieser wurde folgendes notiert:

```
/* Modifikationszeitpunkt: 08.04.2016 - 12:45:56 */
UPDATE 'Ist_In' SET SgNr=2, Austritt=01.05.16 WHERE MSgNr=1;
```

Wenig später ändert Tim Dannys Zuweisung ebenfalls im archivierten Verwaltungsjahr 2015. Dazu spezifiziert Tim dieses zunächst in der Verwaltungsanwendung. Daraufhin wird die Log-Datei *2015.sql* wiederhergestellt. Das bedeutet, dass jede darin enthaltene SQL-Anweisung von der Ältesten bis zur Jüngsten sukzessive vom DBMS ausgeführt werden. Im Anschluss daran wird die von Tim verursachte UPDATE-Anweisung, mit der Dannys Zuweisung zu Sportgruppe „SGKF“ geändert wird, in der Log-Datei *2015.sql* protokolliert. Diese ist nun wie folgt strukturiert:

```
/* Modifikationszeitpunkt: 25.05.2015 - 12:49:34 */
INSERT INTO 'Mitglied' VALUES ('Danny', 'B');

/* Modifikationszeitpunkt: 25.05.2015 - 12:49:34 */
INSERT INTO 'Ist_in' VALUES (1, 1, 1, 01.07.15, '-');

/* Modifikationszeitpunkt: 08.04.2016 - 12:47:12 */
UPDATE 'Ist_In' SET SgNr=2 WHERE MSgNr=1;
```

Wie erkennbar, wurde die UPDATE-Anweisung hinter die anderen, im Verwaltungsjahr 2015 ausgeführten, SQL-Anweisungen angefügt. Beim nächsten Wiederherstellen dieser Datei wird diese Anweisung nach allen anderen ausgeführt. Dadurch hat diese Modifikation solange im Verwaltungsjahr bestand, bis eine weitere SQL-Anweisung bezüglich dieses Datensatzes sie logisch überschreibt. Damit wird illustriert wie das Verändern eines archivierten Datenbestandes im Kontext der datenbankexternen Methode funktioniert.

Außerdem wird durch das nachfolgende Anfügen der SQL-Anweisungen, sowie der zeitlichen Indexierung mit Hilfe der Modifikationszeit die Unveränderbarkeit eines Verwaltungsjahres erreicht. Denn zum einen erkennt man jede Änderung die in einem Verwaltungsjahr getätigt wurde. Dabei sieht man die ursprüngliche Version eines Datensatzes - in diesem Fall die mittlere INSERT-Anweisung - sowie alle nachfolgenden Versionen. Im Kontext dieses Beispiels ist dies die von Tim getätigte UPDATE-Anweisung. Andererseits sieht man exakt wann eine SQL-Anweisung protokolliert wurde. Dadurch wird ersichtlich ob Modifikationen im Verwaltungsjahr getätigt wurden, als dieses operativ war oder erst nachträglich. Vergleicht man die Modifikationszeit der UPDATE-Anweisung mit denen der anderen SQL-Anweisungen, erkennt man, dass diese erst sehr viel später im Jahr 2016 ausgeführt wurde.

Dadurch ist der Verein letztlich in der Lage, beim Blick in die Log-Datei *2015.sql*, Tims nachträgliche Manipulationen aufzuspüren.

4.4 Zusammenfassung

In diesem Kapitel habe ich die datenbankinterne und datenbankexterne Methode vorgestellt. Beide Archivierungsmethoden wurden speziell daran angepasst den Widerspruch aus der vom VGBS verlangten Veränderbarkeit sowie der gesetzlich geforderten Unveränderbarkeit archivierter Daten zu lösen.

Betrachtet man hingegen die weiteren Anforderungen, die an das digitale Archiv des Vereins gestellt werden, ist es nicht trivial zu sagen, welche Methode diese Anforderungen letztlich besser erfüllt.

Einerseits könnte die datenbankinterne Methode besonders gut sein im Hinblick auf den direkten Zugriff auf archivierte sowie protokollierte Informationsobjekte. Da sich diese zusammen mit den operativen Daten in einer Datenbank befinden, kann der Zugriff direkt mittels SQL-Anweisungen geschehen. Dies ist auch der Grund, warum die datenbankinterne Methode vielleicht etwas besser in Bezug auf die Unveränderbarkeit sein könnte. Jede protokollierte Modifikation kann ohne weiteres in der Anwendung mittels SQL-Anweisungen ausgewertet werden.

Andererseits hätte die datenbankexterne Methode wohl den Vorteil, wenn es darum geht die archivierten Daten unabhängig von ihrem Erzeugersystem zu lagern und zu verwalten. Der Grund hierfür ist, dass die archivierten Log-Dateien auf Grund des SQL-Standards theoretisch auch mit jedem beliebigen relationalen DBMS ausgeführt und verwaltet werden können. Außerdem könnte die datenbankexterne Methode besser sein, wenn es darum geht archivierte Daten mit Metadaten zu versehen. Im Kontext der datenbankinternen Methode wäre dies wohl nicht ohne Erweiterung des Datenbankschemas möglich.

Wie ersichtlich haben beide Archivierungsmethoden ihre Vor- und Nachteile, wenn es darum geht die an sie gestellten Anforderungen zu erfüllen. Dadurch ist eine Entscheidung hinsichtlich einer Methode nicht ohne weiteres möglich.

Aus diesem Grund wird ein systematischer tiefgehender Vergleich erforderlich der konkret belegt, welche Archivierungsmethode die an sie gestellten Anforderungen besser erfüllen kann.

5. Evaluation

In diesem Kapitel werde ich den Analytic Hierarchy Process aus [Abschnitt 3.3](#) fortführen und das Entscheidungsproblem dieser Arbeit lösen. Dies bedeutet, ich werde die beiden Alternativen, also die datenbankinterne sowie datenbankexterne Methode, bezüglich der in [Kapitel 3](#) systematisierten Anforderungen vergleichen und bewerten. Die Bewertung erfolgt hierbei danach wie gut eine Anforderung von den beiden Archivierungsmethoden bereits erfüllt wird. Das Ergebnis dieses Kapitels wird ein Index sein. Dieser zeigt konkret in Zahlen ausgedrückt, welche Archivierungsmethode besser im Hinblick auf die Anforderungen sind. Anhand dieses Index, sowie den zuvor erfolgten Vergleichen bzw. Bewertungen, werde ich mich dann für eine Methode entscheiden.

5.1 Vergleich

Die Vergleiche werden in der Reihenfolge durchgeführt in der sie in [Abbildung 5.1](#) dargestellt sind. Das heißt ich werde mit A2 „Direkter Zugriff“ beginnen, wohingegen V4 „Löschen“ die letzte Anforderung sein wird, bezüglich derer die datenbankinterne und datenbankexterne Methode verglichen werden. Jedem Vergleich beigelegt ist eine Tabelle die das Ergebnis des Vergleichs illustriert. Die erste Spalte dieser Tabelle beschreibt, welche der beiden Archivierungsmethoden die jeweilige Anforderung besser erfüllen kann. Die zweite Spalte illustriert die erteilte Bewertung. Die Bewertung erfolgt hierbei mittels der Bewertungsskala des AHP. In diesem Kontext beschreiben dessen Stufen jedoch nicht wieviel wichtiger eine Alternative ist, sondern wieviel besser sie eine Anforderung erfüllen. Zum besseren Verständnis können in [Abbildung 5.1](#) außerdem die Prioritäten jeder einzelnen Anforderung betrachtet werden.

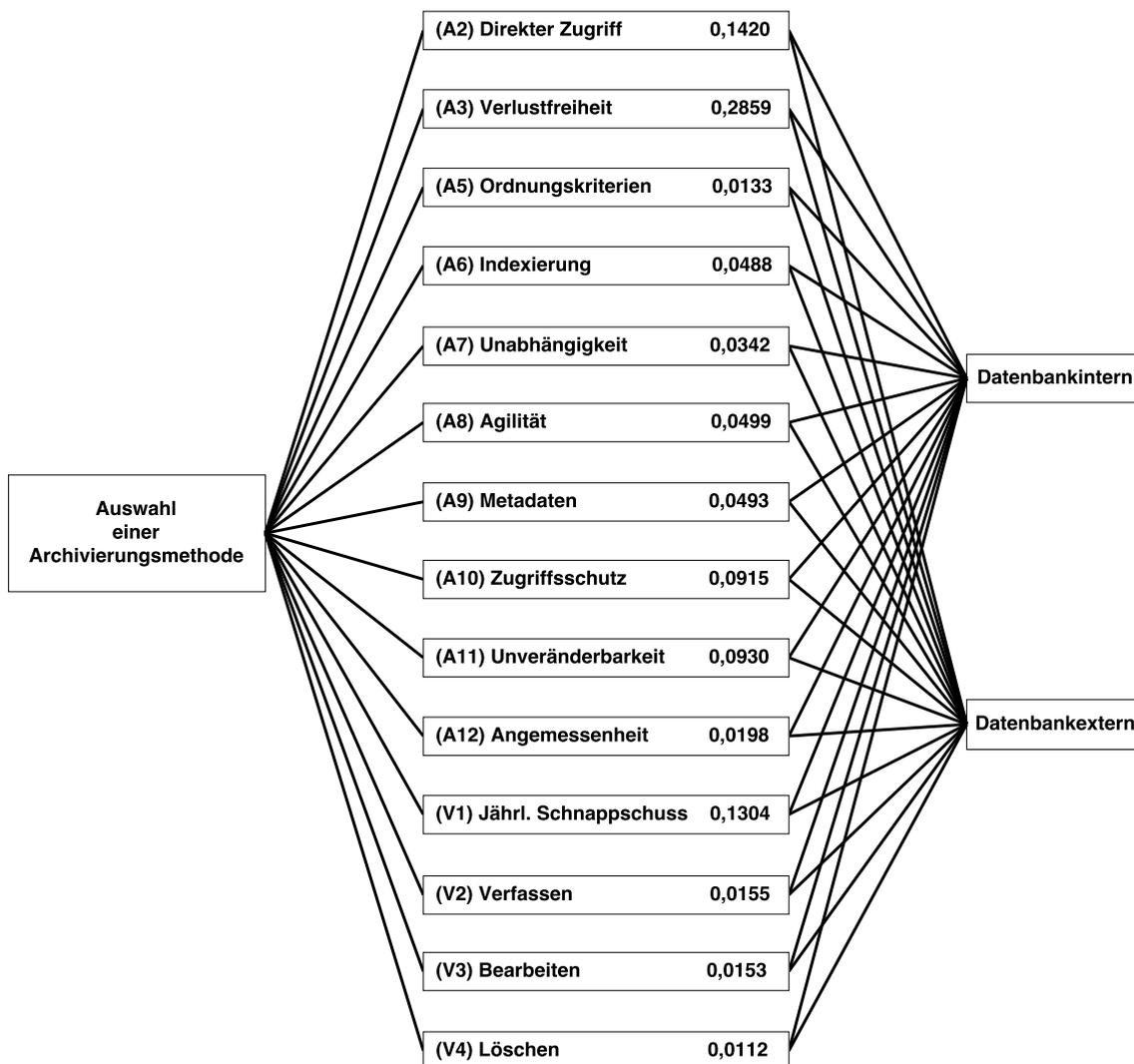


Abbildung 5.1: Hierarchie

5.1.1 (A2) Direkter Zugriff

Im Zusammenhang mit der datenbankinternen Methode befinden sich sämtliche archivierte Daten sowie protokollierte Modifikationen innerhalb des DBS. Dadurch können mit Hilfe von SQL-Anweisungen direkt auf einzelne archivierte sowie protokollierte Informationsobjekte wie z.B. Tupel oder sogar Attribute dieser Daten zugegriffen werden.

Im Rahmen der datenbankexternen Methode sieht dies anders aus: Bevor hierbei die gleiche Funktionalität der datenbankinternen erreicht wird, muss zunächst ein gesamtes Verwaltungsjahr wiederhergestellt werden. Das heißt es müssen alle archivierte Daten eines Jahres wiederhergestellt werden, um auf einen Datensatz zugreifen zu können. Außerdem kann auf protokollierte Modifikationen nicht ohne weiteres über eine Anwendung zugegriffen werden, weil diese lediglich in Form SQL-Anweisungen vorliegen. Somit wäre es erforderlich, dass der Nutzer diese manuell aus einer Log-Datei abliest.

Aus diesen Gründen vergebe ich bezüglich des Vergleichs im Kontext der Anforderung „Direkter Zugriff“ einen Wert von 6 an die datenbankinterne Methode (siehe Tabelle 5.1).

Besser	Bewertung
Intern	6

Tabelle 5.1: Ergebnis des Vergleichs bzgl. „Direkter Zugriff“

5.1.2 (A3) Verlustfreiheit

Prinzipiell erlauben natürlich beide Archivierungsmethoden uneingeschränkt das verlustfreie wiederherstellen archivierter Daten. Eine Archivierungsmethode zu kopieren die dies nicht grundlegend ermöglicht wäre nicht zweckdienlich. Aus diesem Grund wird es im Kontext dieses Vergleichs erforderlich die Wahrscheinlichkeiten zu betrachten, dass keine verlustfreie Wiederherstellung archivierter Daten möglich sein kann.

Die archivierten Daten befinden sich im Rahmen der datenbankinternen Methode innerhalb des DBS. Damit besitzt diese Methode sämtliche Vorzüge die ein modernes DBS in Bezug auf das verlustfreie wiederherstellen von Daten bieten kann. Theoretisch möglich sind zwar Datenverluste beispielsweise durch Manipulationen des Speichers durch einen anderen Prozess oder Abfangen der Datenpakete bei einem Server/Client Setup, jedoch ist dies als recht unwahrscheinlich anzusehen.

Im Gegensatz dazu werden die archivierten Log-Dateien im Kontext der datenbankexternen Methode im wesentlichen vom Nutzer verwaltet. Dabei kann es wiederum passieren, dass eine wiederherzustellende Log-Datei fehlerhaft ist und sich dementsprechend nicht ausführen lässt. Außerdem können Log-Dateien im Laufe ihrer 10 jährigen Aufbewahrungsfrist (siehe Abschnitt 3.1.1.2 „Aufbewahrungsfristen“) verloren gehen. In solchen Fällen wären die darin gespeicherten, sowie alle zeitlich nachfolgenden Daten praktisch verloren. Der Grund hierfür ist, dass jede Log-Datei jeweils nur die Änderungen zur zeitlich vorhergehenden beschreibt, weshalb alle Log-Dateien miteinander verknüpft sind.

Deshalb erhält die datenbankinterne Methode den Wert 3, bezüglich der Anforderung „Verlustfreiheit“ (siehe Tabelle 5.2).

Besser	Bewertung
Intern	3

Tabelle 5.2: Ergebnis des Vergleichs bzgl. „Verlustfreiheit“

5.1.3 (A5) Ordnungskriterien

Da sich die archivierten Daten im Kontext der datenbankinternen Methode innerhalb einer relationalen Datenbank befinden, können diese nach jedem beliebigen Ordnungskriterium ausgewertet werden, das diese bietet. Dies schließt beispielsweise das aufsteigende bzw. absteigende sortieren mittels ORDER BY-Anweisung nach

einem bestimmten Attribut ein, aber auch das Selektieren nur derer Datensätze die eine bestimmte Bedingung erfüllen. Diese Ordnungskriterien können hierbei auf sämtliche Daten angewendet werden. Das heißt auch auf Datensätze aus beliebigen Verwaltungsjahren oder verschiedenen Versionen einzelner Datensätze. So können Datensätze aus unterschiedlichen Verwaltungsjahren bzw. unterschiedliche Versionen dieser dem Nutzer nach einem Kriterium geordnet bereitgestellt werden.

Entscheidet sich der Nutzer im Rahmen der datenbankexternen Methode ein Verwaltungsjahr wiederherzustellen, so befinden sich in Folge dessen die archivierten Daten dieses Jahres ebenfalls in der Datenbank. Dort können die oben besprochenen Ordnungskriterien prinzipiell auch in diesem Kontext angewendet werden. Einzig das geordnete bereitstellen archivierter Daten aus verschiedenen Verwaltungsjahren funktioniert nicht, weil jedes Verwaltungsjahr nur für sich alleine ausgewertet werden kann. Zudem können keine Ordnungskriterien auf protokollierte Modifikationen angewendet werden, weil dies die Reihenfolge der SQL-Anweisungen in einer Log-Datei durcheinander bringen würde. Dadurch könnte eine Log-Datei unter Umständen auf Grund von Integritätsverletzungen und ähnlicher Gründe nicht mehr ausführbar sein.

Auf Grundlage dieser Ausführungen vergebe ich den Wert 4 an die datenbankinterne Methode (siehe [Tabelle 5.3](#)).

Besser	Bewertung
Intern	4

Tabelle 5.3: Ergebnis des Vergleichs bzgl. „Ordnungskriterien“

5.1.4 (A6) Indexierung

Im Zusammenhang mit der datenbankinternen Methode existieren im wesentlichen nur zwei Möglichkeiten einen Datensatz eindeutig zu indexieren (siehe [Abschnitt 2.2.2.2](#) „Primärschlüssel“): Einerseits durch einem Verbund aus Objekt-Schlüssel, einem Attribut der VGBS-Gültigkeitszeit sowie der Transaktionszeit. Andererseits mit Hilfe eines Surrogatschlüssels.

Dahingegen besitzt der Datenbankdesigner im Kontext der datenbankexternen Methode sämtliche Freiheiten bei der Auswahl eines Schlüssels und muss keine zeitlichen Dimensionen beachten.

Beide Archivierungsmethoden erreichen dadurch das eindeutige Indexieren eines operativen bzw. archivierten Datensatzes. Deswegen sind sowohl die datenbankinterne, als auch die datenbankexterne Methode bezüglich der Anforderung „Indexierung“ gleichwertig (siehe [Tabelle 5.4](#)).

Besser	Bewertung
Beide	1

Tabelle 5.4: Ergebnis des Vergleichs bzgl. „Indexierung“

5.1.5 (A7) Unabhängigkeit

Die Anforderung „Unabhängigkeit“ besagt, dass das Archiv seine Daten unabhängig von ihrem Erzeugersystem lagern, verwalten und bereitstellen soll (siehe [Abschnitt 3.2.1.7 „Unabhängigkeit“](#)). Im Zusammenhang mit der datenbankinternen und datenbankexternen Methode besteht das Erzeugersystem im Kern jeweils aus der Anwendung und dem DBS. Diese bilden hierbei auch das Archiv, wobei zu diesem auch die archivierten Daten gehören.

Das Archiv im Kontext der datenbankinternen Methode kann seine gespeicherten Daten nicht unabhängig von ihrem Erzeugersystem lagern, verwalten bzw. bereitstellen. Der Grund hierfür ist zum einen, dass die archivierten Daten direkt in das DBS integriert sind. Somit können diese nur darin gelagert sowie verwaltet werden. Zum anderen wird die Anwendung zur Umsetzung der temporalen Konzepte benötigt. Diese werden von keinem herkömmlichen DBS vollständig unterstützt.

Im Gegensatz dazu werden im Rahmen der datenbankexternen Methode die archivierten Daten unabhängig von ihrem Erzeugersystem gelagert. Das heißt diese befinden sich in einem Speicherort der unabhängig von dem der Anwendung bzw. dem DBS ist. Das Verwalten sowie bereitstellen kann hierbei ebenfalls mit DBS eigenen Mitteln erfolgen. Dadurch sind archivierte Daten hierbei im wesentlichen auch nicht abhängig von der Anwendung. Einzig die Abhängigkeit vom DBS bleibt bestehen, weil es sich bei den archivierten Daten um SQL-Dateien handelt. Die darin gespeicherten Daten können nur verwaltet werden, wenn die entsprechenden Log-Dateien vom DBS ausgeführt werden.

Basierend darauf vergebe ich den Wert 7 an die datenbankexterne Methode (siehe [Tabelle 5.5](#)).

Besser	Bewertung
Extern	7

Tabelle 5.5: Ergebnis des Vergleichs bzgl. „Unabhängigkeit“

5.1.6 (A8) Agilität

Die Agilität eines Archivs meint dessen Fähigkeit auf Schemaevolutionen reagieren zu können, um Datenverluste zu vermeiden. Arten von Schemaevolutionen können dabei sein, dass neue Schemaelemente hinzukommen oder bestehende bearbeitet bzw. gelöscht werden.

Bei der datenbankinternen Methode befinden sich die archivierten zusammen mit den operativen Daten in einer Datenbank. Dadurch führen nur das Bearbeiten bzw. Löschen von Schemaelementen zum Verlust dieser Daten. Wird eine Tabelle gelöscht, so werden sämtliche darin enthaltenen Daten gelöscht. Das Löschen eines Attributs führt dazu, dass dessen Daten bei allen Datensätzen verloren gehen. Ändert sich der Datentyp oder die Zeichenlänge eines Attributs, so wirkt sich dies ebenfalls lokal auf die Ausprägungen dieses Attributs aus.

Dies gilt prinzipiell auch für die datenbankexterne Methode. Dessen archivierte Daten müssen in die Datenbank überführt werden, bevor der Nutzer sie effektiv verwalten kann. Hierbei können jedoch sämtliche Arten von Schemaevolutionen dazu führen, dass sich die archivierten Daten nicht ohne weiteres wiederherstellen lassen. Wird beispielsweise ein Attribut neu eingeführt, das nicht null sein darf, so können ältere INSERT-Anweisungen, die dieses Attribut nicht beinhalten, nicht ausgeführt werden. Steht solch eine INSERT-Anweisung zu Beginn einer Log-Datei, kann das gesamte Verwaltungsjahr und alle darauf folgenden nicht wiederhergestellt werden. Erst wenn alle dieser Anweisungen manuell um das entsprechende Attribut ergänzt wurden, ist dies wieder möglich.

Zwar ist dies kein direkter Verlust von Daten, stellt jedoch einen großen Nachteil der datenbankexternen Methode dar. Aus diesem Grund erhält die datenbankinterne Methode den Wert 5 bezüglich der Anforderung „Agilität“ (siehe [Tabelle 5.6](#)).

Besser	Bewertung
Intern	5

Tabelle 5.6: Ergebnis des Vergleichs bzgl. „Agilität“

5.1.7 (A9) Metadaten

In beiden Methoden werden archivierte Daten mit Metadaten die ihrer Verwaltung dienen ausgestattet. Im Rahmen der datenbankinternen Methode sind dies zum einen das zeitliche Intervall der VGBS-Gültigkeitszeit das beschreibt, wann ein Datensatz operativ war. Andererseits das Intervall der Transaktionszeit, welches darstellt wann Daten in die Datenbank gespeichert und logisch gelöscht wurden. Beide werden durch Attribute einer Tabelle ausgedrückt.

Bei der datenbankexternen Methoden beschreiben Metadaten, zu welchem Verwaltungsjahr eine Log-Datei gehört und wann eine SQL-Anweisung ausgeführt wurde. Erstere werden hierbei durch den Namen einer Log-Datei repräsentiert.

Sollen weitere Metadaten hinzugefügt werden z.B. der Name des Autors eines Datensatzes, hat die datenbankinterne Methode den Nachteil, dass hierfür stets neue Attribute der Tabelle hinzugefügt werden müssen. Dies macht das Datenbankschema unübersichtlicher, erhöht die Speicherauslastung der operativen Datenbank und kann letztlich dazu führen, dass diese langsamer wird. Dahingegen können im Kontext der datenbankexternen Methode weitere Metadaten schlicht, als Kommentar an jede SQL-Anweisung geschrieben werden. Auch möglich ist eine zusätzliche Datei dafür anzulegen. Diese Maßnahmen erhöhen in erster Linie nur den Speicherverbrauch auf der Festplatte, führen jedoch nicht zu Performance-Verlusten beim Zugriff auf archivierte oder gar operative Daten. Aus diesem Grund erhält die datenbankexterne Methode hierbei den Wert 5 (siehe [Tabelle 5.7](#)).

Besser	Bewertung
Extern	5

Tabelle 5.7: Ergebnis des Vergleichs bzgl. „Metadaten“

5.1.8 (A10) Zugriffsschutz

Lässt man die Möglichkeiten der Anwendung außer acht, so besteht bei beiden Methoden ein gewisser Zugriffsschutz darin, dass die Datenbank mit einem Passwort gesichert ist.

Besitzt man jedoch im Rahmen der datenbankinternen Methode Zugriff auf die operativen Daten, so erhält man diesen auch auf sämtliche archivierten Daten. Der Grund hierfür ist, dass diese zusammen in einer Datenbank gehalten werden.

Im Gegensatz dazu sind die archivierten Daten im Kontext der datenbankexternen Methode getrennt von den operativen Daten. Erstere befinden sich dabei in einem passwortgeschützten Zip-Archiv. Aus diesem Grund wird zusätzlich zum Passwort für die Datenbank, eines zur Wiederherstellung der archivierten Daten erforderlich.

Dieser doppelte Zugriffsschutz ist der Grund, weshalb die datenbankexterne Methode den Wert 5 bezüglich der Anforderung „Zugriffsschutz erhält (siehe [Tabelle 5.8](#)).

Besser	Bewertung
Extern	5

Tabelle 5.8: Ergebnis des Vergleichs bzgl. „Zugriffsschutz“

5.1.9 (A11) Unveränderbarkeit

Die datenbankinterne Methode gewährleistet die Unveränderbarkeit archivierter Daten mit Hilfe der Transaktionszeit eines Datensatzes (siehe [Abschnitt 2.2.1.10](#)). Diese ist direkt mit dessen Datenmodifikationen verbunden und bildet diese sekunden genau ab. Dabei wird kein Datensatz mehr gelöscht oder überschrieben. Stattdessen wird entweder das Ende der Transaktionszeit eines zu löschenden Datensatzes gesetzt oder eine modifizierte Version eines Datensatzes hinzugefügt, wenn dieser überschrieben werden soll.

Durch das zeitlich indexierte Protokollieren von modifizierenden SQL-Anweisungen stellt die datenbankexterne Methode die Unveränderbarkeit sicher.

Somit erfüllen beide Archivierungsmethoden diese Anforderung. Die datenbankinterne erhält jedoch den Wert 5, weil auf die einzelnen Versionen eines Datensatzes mittels SQL-Anweisungen zugegriffen werden kann. Dadurch können diese ohne weiteres aus der Anwendung heraus ausgewertet werden. Dies ist besonders im Hinblick auf die Auswertung durch den Gesetzgeber von Vorteil.

Dahingegen können die protokollierten SQL-Anweisungen im Kontext der datenbankexternen Methode nur in den entsprechenden Log-Dateien ausgewertet werden. Hierbei ist zum einen zunächst ein Verständnis von SQL gefordert. Andererseits beschreibt eine UPDATE-Anweisung immer nur eine Veränderung eines Datensatzes und nicht dessen gesamte Struktur. Dadurch wird es erforderlich jede SQL-Anweisung die zu einem Datensatz gehört aufzufinden, um sich ein ganzheitliches Bild machen zu können. In diesem Kontext referieren UPDATE-Anweisungen oft nur mit einem numerischen Schlüssel auf einen Datensatz und können sich verstreut über viele Log-Dateien befinden. Diese Gegebenheiten können die Auswertung von protokollierten Modifikationen beträchtlich erschweren.

Besser	Bewertung
Intern	5

Tabelle 5.9: Ergebnis des Vergleichs bzgl. „Unveränderbarkeit“

5.1.10 (A12) Angemessenheit

Im Zusammenhang mit der datenbankinternen Methode können archivierte Daten mit der Geschwindigkeit eines DBMS bereitgestellt werden. Einzig dadurch, dass sämtliche archivierten Daten innerhalb der Datenbank verweilen, kann über viele Jahre hinweg zu Geschwindigkeitseinbußen führen.

Bevor diese Vorteile auch den archivierten Daten der datenbankexternen Methode zu gute kommen, müssen zunächst eine bestimmte Anzahl an Log-Dateien wiederhergestellt werden. Die gesetzliche Aufbewahrungspflicht für die Daten des VGBS beträgt 10 Jahre (siehe [Abschnitt 3.1.1.2](#) „Aufbewahrungsfristen“). Wurden über diesen Zeitraum jeden Werktag SQL-Anweisungen protokolliert, so kann es unter Umständen sehr lange dauern bis ein bestimmtes Verwaltungsjahr wiederhergestellt wurde. Dies ist insbesondere der Fall, wenn auf Daten aus dem vorhergehenden Verwaltungsjahr zugegriffen werden soll. Dann müssen insgesamt 9 (eine für jedes Verwaltungsjahr) Log-Dateien wiederhergestellt werden.

Aus diesem Grund vergebe ich im Rahmen dieses Vergleichs den Wert 8 an die datenbankinterne Methode (siehe [Tabelle 5.10](#)).

Besser	Bewertung
Intern	8

Tabelle 5.10: Ergebnis des Vergleichs bzgl. „Angemessenheit“

5.1.11 (V1-4) VGBS-Anforderungen

Die Anforderungen des VGBS umfassen das Archivieren eines jährlichen Schnappschuss der Vereinsdaten, sowie das Verfassen, Bearbeiten und Löschen von Daten eines archivierten Datenbestandes. Sowohl die datenbankinterne, als auch die datenbankexterne Methode wurden speziell für diese Anforderungen konzipiert. Dementsprechend erfüllen beide Archivierungsmethoden die Anforderungen des Vereins gleichwertig. Wie dies erfolgt kann in [Kapitel 4](#) sowie im Anhang A in [Abschnitt A.2](#) „Funktionsweise der Archivierungsmethoden“ nachgeschlagen werden. Aus diesem Grund erhalten die datenbankinterne und datenbankexterne Methode hierbei den Wert 1.

Anforderung	Besser	Bewertung
Jährl. Schnappschuss	Beide	1
Verfassen	Beide	1
Bearbeiten	Beide	1
Löschen	Beide	1

Tabelle 5.11: Ergebnis des Vergleichs bzgl. der Anforderungen des VGBS

5.2 Entscheidung

In [Tabelle 5.12](#) kann der Index, also das Ergebnis des Vergleichs sowie die gewichteten Bewertungen betrachtet werden. Hierbei ist deutlich zu erkennen, dass die datenbankinterne Methode mit 0,622 die beste Methode ist, wenn es darum geht die Anforderungen an das digitale Archiv des VGBS zu erfüllen. Dahingegen wurde die datenbankexterne Methode insgesamt mit 0,378 gewertet. Der Grund dafür ist, dass die datenbankinterne Methode in fast der Hälfte der Vergleiche besser abgeschnitten hat. Darunter befinden sich besonders hoch priorisierte Anforderungen wie das verlustfreie Wiederherstellen archivierter Daten, aber auch der direkte Zugriff auf diese (siehe [Abbildung 5.1](#)). Im Gegensatz dazu kann die datenbankexterne letztlich nur drei Anforderungen besser erfüllen: „Zugriffsschutz“, „Metadaten“ und „Unabhängigkeit“. Hiervon ist nur „Zugriffsschutz“ mit 0,0915 eine höher priorisierte Anforderung. Auf Grundlage dieser Daten entscheide ich mich deswegen dafür, die datenbankinterne Methode im Kontext der Verwaltungsanwendung des VGBS umzusetzen.

Methode	A2	A3	A5	A6	A7	A8	A9	A10	A11	A12	V1	V2	V3	V4	Erg.
intern	0,122	0,214	0,011	0,024	0,004	0,042	0,008	0,015	0,078	0,018	0,065	0,008	0,008	0,006	0,622
extern	0,020	0,071	0,003	0,024	0,030	0,008	0,041	0,076	0,016	0,002	0,065	0,008	0,008	0,006	0,378

Tabelle 5.12: Übersicht der gewichteten Bewertungen sowie des Ergebnisses des Vergleichs

6. Schlussbetrachtungen

Der nachfolgende Abschnitt ist wie folgt untergliedert: Zunächst werde ich eine Zusammenfassung der Ergebnisse dieser Arbeit präsentieren. Abschließend werde ich einen Ausblick bezüglich zukünftiger Optimierungen sowie Vergleiche der datenbankinternen und datenbankexternen Methode geben.

6.1 Zusammenfassung

Das primäre Ziel dieser Arbeit war es ein Konzept zur Archivierung der Daten des VGBS zu entwickeln. Dazu habe ich zunächst die Anforderungen analysiert sowie priorisiert, die an das digitale Archiv des Vereins gestellt werden. Diese ließen sich in zwei Kategorien untergliedern:

Zum einen in allgemeine Anforderungen, welche von jedem digitalen Archiv erfüllt werden sollten. Diese habe ich aus Fachliteratur, sowie aus den gesetzlichen Vorgaben zur digitalen Archivierung hergeleitet. Auf der anderen Seite in jene Anforderungen, welche direkt vom VGBS gestellt werden. Diese resultieren in erster Linie aus den Bedürfnissen der bisherigen Archivierung des Vereins.

Die Analyse der Anforderungen hat gezeigt, dass ein Widerspruch zwischen den allgemeinen Anforderungen und denen des VGBS besteht. Hierbei wurde im Kontext der allgemeinen Anforderungen die vom Gesetzgeber geforderte Unveränderbarkeit archivierter Daten identifiziert. Diese Anforderung besagt, dass der Ursprungszustand, sowie sämtliche Modifikationen archivierter Daten nachvollziehbar sein müssen. Außerdem muss Kennlich sein, ob eine Modifikation nachträglich oder ursprünglich geschehen ist.

Dem gegenüber steht die Anforderung des VGBS seine archivierten Daten nachträglich umfassend Modifizieren zu können. Die Modifikationen beinhalten dabei das Verfassen, Bearbeiten, sowie Löschen von Daten eines archivierten Datenbestandes. Diese Funktionen benötigt der Verein, um die archivierten Daten an die aus seiner Sicht reellen Gegebenheiten anzupassen.

Die Schlussfolgerung daraus war, dass nur eine Archivierungslösung in Frage kommt die Änderungen an archivierten Daten zulässt, diese jedoch protokolliert. Nur dadurch kann der Widerspruch aus der vom Verein geforderten Veränderbarkeit sowie der gesetzlich vorgegebenen Unveränderbarkeit gelöst werden.

Ferner hat die vorliegende Arbeit gezeigt, dass im wesentlichen zwei Ansätze existieren um die Daten des Vereins zu archivieren: Einmal den datenbankinternen Ansatz. Im Kontext dessen werden die archivierten Daten innerhalb des DBS gelagert. Andererseits den datenbankexternen Ansatz, bei dem die archivierten Daten in Dateien außerhalb des DBS aufbewahrt werden.

Keine dieser Ansätze erfüllte für sich alleine die oben beschriebene Funktionalität Daten zu archivieren und parallel dazu Modifikationen an diesen Daten zu protokollieren. Aus diesem Grund war es erforderlich darauf aufbauend Archivierungsmethoden zu konzipieren, welche dies ermöglichen und somit den oben beschriebenen Widerspruch lösen.

Daraus resultierend habe ich die datenbankinterne sowie die datenbankexterne Methode entwickelt. Dafür habe ich die bereits erwähnten Ansätze angepasst, sodass beide Archivierungsmethoden den Widerspruch aus Unveränderbarkeit und Veränderbarkeit lösen können. Somit erfüllen beide Methoden die Anforderungen, welche an das digitale Archiv des VGBS gestellt werden. Im Zusammenhang damit war es jedoch nicht trivial zu entscheiden, welche Methode die gegebenen Anforderungen qualitativ besser erfüllt, sodass eine Entscheidung hinsichtlich einer Methode möglich gewesen wäre.

Aus diesem Grund habe ich bezüglich der Anforderungen einen qualitativen Vergleich der beiden Archivierungsmethoden durchgeführt. Dieser hat gezeigt, dass die datenbankinterne Methode mit einem Wert von 0,662 die gestellten Anforderungen am besten erfüllen kann. Dahingegen erhielt die datenbankexterne Methode den Wert 0,378. Damit wird die datenbankinterne Methode im Kontext der Verwaltungsanwendung des VGBS umgesetzt.

6.2 Ausblick

Der Vergleich hat gezeigt, dass die datenbankinterne Methode die bessere ist im Hinblick auf die Anforderungen, die an beide Methoden gestellt werden. Momentan handelt es sich hierbei noch um einen rein konzeptionellen und theoretischen Vergleich. Außerdem basiert dieser primär auf meinen Erfahrungen beim ersten prototypischen umsetzen beider Archivierungsmethoden.

Weiterhin hat der Vergleich verschiedene Schwachstellen beider Methoden herausgestellt. Dies betrifft insbesondere die datenbankexterne Methode. Momentan ist es hierbei noch erforderlich eine bestimmte Anzahl an Log-Dateien wiederherzustellen, bevor ein Verwaltungsjahr betrachtet werden kann. Dies kann sehr viel Zeit in Anspruch nehmen. Außerdem ist jede Log-Datei von der zeitlich vorhergehenden abhängig. Wäre hierbei eine Log-Datei nicht verfügbar, so könnten alle zeitlich darauffolgenden nicht wiederhergestellt werden. Vorstellbar ist es deswegen zusätzlich zu den Log-Dateien, von jedem Verwaltungsjahr einen Schnappschuss anzufertigen.

Dann müsste nur noch dieser wiederhergestellt werden, um das Verwaltungsjahr betrachten zu können.

Diese Optimierung hätte unter Umständen Auswirkungen auf die Bewertung bezüglich den Anforderungen *Direkter Zugriff*, *Verlustfreiheit* sowie *Angemessenheit*. Davon sind die ersten beiden besonders hoch priorisierte Anforderungen. Um den Einfluss dieser Optimierung auf das Gesamtergebnis abschätzen zu können, wäre somit zukünftig ein weiterer Vergleich denkbar. Dieser könnte zusätzlich dazu unter Betrachtung gewisser quantitativer Abschätzungen bezüglich Performance und Speicherverbrauch beider Methoden erfolgen. In diesem Kontext wäre auch eine Optimierung der datenbankinternen Methode vorstellbar, bei der nicht mehr aktuelle Datensätze in eine separate Tabelle ausgelagert werden. Somit könnten bei einer großen Menge an Daten insbesondere die Zugriffszeiten auf operative, sowie archivierte verringert werden.

A. Anhang

A.1 Vergleich der Anforderungen

A.1.1 (A2) Direkter Zugriff

(A3) Verlustfreiheit: Ein Archiv, das seine Daten nicht verlustfrei Wiederherstellen kann erfüllt keinen Zweck, weil die Authentizität dieser Daten im Falle einer gesetzlichen Überprüfung dadurch nicht gewährleistet werden kann. Deswegen ist Verlustfreiheit extrem viel wichtiger.

Bewertung: 1/9

(A5) Ordnungskriterien: Der direkte Zugriff auf einzelne Informationsobjekte ist sehr viel wichtiger, weil dieser für die Auswertung im Rahmen einer steuerrechtlichen Außenprüfung essentiell ist. Hätte der Gesetzgeber hierbei nicht die Möglichkeit Daten einzelner Tabellen auszuwerten, so könnte eine Außenprüfung nicht im vollem Umfang stattfinden. Dies hätte unter Umständen rechtliche Konsequenzen für den VGBS. Dahingegen können Informationsobjekte, auf die direkt zugegriffen werden kann, auch ohne Ordnungskriterien früher oder später wiedergefunden werden.

Bewertung: 8

(A6) Indexierung: Der direkte Zugriff auf einzelne Informationsobjekte ist sehr viel wichtiger, weil dieser für die Auswertung im Rahmen einer steuerrechtlichen Außenprüfung essentiell ist. Hätte der Gesetzgeber hierbei nicht die Möglichkeit beispielsweise Daten einzelner Tabellen auszuwerten, so könnte eine Außenprüfung nicht im vollem Umfang stattfinden. Dies hätte unter Umständen rechtliche Konsequenzen für den VGBS. Dahingegen können Informationsobjekte, auf die direkt zugegriffen werden kann, auch einen eindeutigen Index früher oder später wiedergefunden werden.

Bewertung: 8

(A7) Unabhängigkeit / (A8) Agilität: Ein Archiv, das seine Daten unabhängig von ihrem Erzeugersystem verwalten sowie bereitstellen kann bzw. agil gegenüber strukturellen organisatorischen Änderungen ist und somit Informationsverlusten vorbeugt, ist nicht Sinnvoll, wenn nicht auch sämtliche in diesem Archiv enthaltenen Informationen ausgewertet werden können, welches der Hauptzweck der Archivierung ist. Aus diesem Grund ist ein direkter Zugriff viel wichtiger.

Bewertung: 7

(A9) Metadaten: Der direkte Zugriff auf einzelne Informationsobjekte ist viel wichtiger, weil dieser für die Auswertung im Rahmen einer steuerrechtlichen Außenprüfung essentiell ist. Hätte der Gesetzgeber hierbei nicht die Möglichkeit beispielsweise Daten einzelner Tabellen auszuwerten, so könnte eine Außenprüfung nicht im vollem Umfang stattfinden. Dies hätte unter Umständen rechtliche Konsequenzen für den VGBS. Dahingegen beschreiben Metadaten diese Informationsobjekte und Helfen dabei diese zu verstehen. Dies ist jedoch nicht sinnvoll, wenn auf die eigentlichen Daten eines Informationsobjekt nicht zugegriffen werden kann.

Bewertung: 7

(A10) Zugriffsschutz / (A11) Unveränderbarkeit: Informationen zu schützen, auf die nicht im vollem Umfang zugegriffen werden kann, erfüllt nur einen geringen Zweck. Deswegen ist der direkte Zugriff viel wichtiger.

Bewertung: 7

(A12) Angemessenheit: Das schnelle Bereitstellen einzelner archivierter Informationsobjekte auf die letztlich nicht direkt zugegriffen werden kann, erfüllt nur einen geringen Zweck. Aus diesem Grund ist der direkte Zugriff sehr viel wichtiger.

Bewertung: 8

(V1) Jährl. Schnappsch.: Beide Anforderungen sind gleichwertig, weil die Informationsobjekte die man archiviert damit einhergehen, dass man auf diese auch im vollem Umfang zugreifen kann.

Bewertung: 1

(V2) Verfassen: Der direkte Zugriff auf einzelne Informationsobjekte ist viel wichtiger, weil dieser für die Auswertung im Rahmen einer steuerrechtlichen Außenprüfung essentiell ist. Hätte der Gesetzgeber hierbei nicht die Möglichkeit beispielsweise Daten einzelner Tabellen auszuwerten, so könnte eine Außenprüfung nicht im vollem Umfang stattfinden. Dies hätte unter Umständen rechtliche Konsequenzen für den VGBS.

Bewertung: 7

(V3) Bearbeiten / (V4) Löschen: Der direkte Zugriff auf einzelne Informationsobjekte ist extrem viel wichtiger, weil das Bearbeiten dieser Informationsobjekte nicht ohne diesen möglich wäre.

Bewertung: 9

A.1.2 (A3) Verlustfreiheit

(A5) Ordnungskriterien: Ordnungskriterien sind wichtig um archivierte Daten effizient wiederfinden zu können. Diese besitzen jedoch nur wenig Zweck, wenn die wiedergefundenen Daten nicht mehr vollständig der Form entsprechen in der sie ursprünglich archiviert wurden. Aus diesem Grund ist die Verlustfreiheit extrem viel wichtiger.

Bewertung: 9

(A6) Indexierung: Auch hierbei gilt, dass das verlustfreie Wiederherstellen archivierter Daten extrem viel wichtiger ist. Der Hintergrund dessen ist der gleiche wie im Kontext der Ordnungskriterien nämlich, dass das eindeutige identifizieren eines Datensatzes schon prinzipiell wichtig ist, jedoch keinen Zweck erfüllt, wenn die identifizierten Daten nicht frei von Verlusten sind.

Bewertung: 9

(A7) Unabhängigkeit: Auch die Unabhängigkeit eines Archivs ist nichts Wert, wenn archivierte Daten im Falle einer steuerrechtlichen Außenprüfung nicht verlustfrei wiederhergestellt werden können.

Bewertung: 9

(A8) Agilität: Für die Agilität eines Archivs gilt die gleiche Begründung wie für die Unabhängigkeit eines Archivs.

Bewertung: 9

(A9) Metadaten: Auch der Vorteil von Metadaten zu verstehen, wie das Erzeugersystem eines archivierten Datensatz beschaffen war, hat keinerlei Bedeutung, wenn dieser sowieso nur unter Verlusten wiederherstellbar ist. Aus diesem Grund ist hierbei ebenfalls die Verlustfreiheit von extremer Wichtigkeit

Bewertung: 9

(A10) Zugriffsschutz / (A11) Unveränderbarkeit: Der Schutz von Daten deren Authentizität nicht sichergestellt werden kann ist sinnlos. Daher ist Verlustfreiheit auch in diesem Zusammenhang extrem wichtig.

Bewertung: 9

(A12) Angemessenheit: Hierbei stellt sich die Frage, warum der Verein archivierte Daten schnell wiederherstellen will, wenn diese sowieso nicht mehr eins zu eins denen von ihm gespeicherten Daten entsprechen können? Deswegen stellt sich auch hier die extrem große Wichtigkeit der Verlustfreiheit heraus.

Bewertung: 9

(V1) Jährl. Schnappsch.: Ein jährlicher Schnappschuss der Vereinsdaten bildet nicht den korrekten Datenbestand eines Jahres ab und somit auch nicht die aufzubewahrenden Aufzeichnungen des VGBS, wenn darin befindliche Daten in Folge des Zugriffs auf diesen Verluste aufweisen. Deswegen ist die Verlustfreiheit auch hierbei extrem viel wichtiger.

Bewertung: 9

(V2) Verfassen / (V3) Bearbeiten: Das Verfassen bzw. Bearbeiten archivierter Daten, die im Nachhinein in Folge eines Zugriffs nicht mehr in der Form existent sein können, wie der VGBS sie verfasst hat ist sinnlos. Vor diesem Hintergrund ist die Verlustfreiheit extrem viel wichtiger als das Verfassen archivierter Daten.

Bewertung: 9

(V4) Löschen: Das Löschen von archivierten Datensätzen dient dem Zweck Daten zu korrigieren bzw. an reelle Gegebenheiten anzupassen. Stellt ein Archiv Daten nicht ohne Verluste wieder her, wäre der VGBS ohne Pause ausschließlich dabei diese zu korrigieren. Aus diesem Grund ist die Verlustfreiheit auch in diesem Kontext extrem viel wichtiger.

Bewertung: 9

A.1.3 (A5) Ordnungskriterien

(A6) Indexierung: Daten die keiner Ordnung unterliegen sind schwer zu finden, können jedoch mit gewissen Aufwand letztlich doch gefunden werden. Dahingegen können Daten die keinen eindeutigen Index aufweisen unter Umständen gar nicht mehr identifiziert werden. Aus diesem Grund ist die Indexierung wichtiger.

Bewertung: 1/4

(A7) Unabhängigkeit: Die Folgen eines obsoleten Erzeugersystems können fatal sein, wenn die archivierte Daten nicht unabhängig von diesem aufbewahrt werden. Hierbei ist im schlimmsten Fall mit dem Verlust aller archivierten Daten zu rechnen. Im Gegensatz dazu ist es eher ein geringes Problem, wenn Daten nicht geordnet sind und deswegen schwieriger wiedergefunden werden. Deswegen ist die Unabhängigkeit viel wichtiger.

Bewertung: 1/7

(A8) Agilität: Ob sich irgendwann die organisatorischen Strukturen des VGBS verändern ist ungewiss und lässt sich nicht abschätzen. Jedoch sind die Folgen fehlender Agilität, dass im schlimmsten Fall nicht mehr auf archivierte Daten zugegriffen werden kann sehr groß. Im Gegensatz dazu ist es eher ein geringes Problem, wenn Daten nicht geordnet sind und deswegen schwieriger wiedergefunden werden. Deswegen ist die Agilität eines Archivs viel wichtiger.

Bewertung: 1/7

(A9) Metadaten: Mit Hilfe von Metadaten können archivierte Daten schnell und effizient wiedergefunden werden. Über dies tragen sie zur Nachhaltigkeit dieser bei. Ordnungskriterien unterstützen zwar auch das Wiederfinden von Daten, jedoch nur in einem begrenzten Rahmen. Mit ihnen ist es nur möglich die Auswahl einzugrenzen und einen Überblick zu verschaffen, wohingegen durch Metadaten einzelne archivierte Daten gezielt gesucht werden können. Aus diesem Grund sind Metadaten wichtiger.

Bewertung: 1/5

(A10) Zugriffsschutz: Der Zugriffsschutz ist viel wichtiger, weil ein Fehlen dessen eine Gefahr für den Verein und seine Kunden darstellt, wenn dessen sensible Daten in falsche Hände geraten. Dahingegen sind zwar die Folgen fehlender Ordnungskriterien, dass sich Daten nicht ohne weiteres auffinden lassen, bei jedem Zugriff auf die archivierten Daten zu spüren, im Vergleich nicht annähernd so verheerend sind.

Bewertung: 1/5

(A11) Unveränderbarkeit: Sowohl die Unveränderbarkeit, als auch Ordnungskriterien sind gesetzlich vorgeschrieben. Hierbei muss die Notwendigkeit nach Ordnungskriterien abgeleitet werden, wohingegen die Unveränderbarkeit konkret geschrieben steht. Der Verein profitiert prinzipiell von beiden, wobei der VGBS noch einen größeren Nutzen davon hat, dass seine Daten im Falle einer gesetzlichen Auswertung Authentisch sind. Könne dies nicht gewährleistet werden drohen rechtliche Konsequenzen. Somit ist die Unveränderbarkeit viel wichtiger.

Bewertung: 1/6

(A12) Angemessenheit: Sowohl die Angemessenheit, als auch Ordnungskriterien sind gesetzlich vorgeschrieben. Hierbei muss die Notwendigkeit von Ordnungskriterien abgeleitet werden, wohingegen die Angemessenheit konkret geschrieben steht. Hierbei gilt jedoch, dass die Angemessenheit sehr unpräzise formuliert ist und der Verein auch etwas länger auf seine archivierten Vereinsdaten warten kann, wenn diese dafür geordnet sind. Daher sind Ordnungskriterien wichtiger.

Bewertung: 4

(V1) Jährl. Schnappsch.: Einen jährlichen Schnappschuss der Vereinsdaten zu besitzen, ist eine Forderung des Vereins, um der Aufbewahrungspflicht seiner Aufzeichnungen nachzukommen. Kann der VGBS dies nicht können hohe rechtliche Strafen die Konsequenz sein. Dahingegen lässt sich die Forderung nach Ordnungskriterien zwar aus §146 Absatz 1 AO ableiten, jedoch ist ein vollständiges Archiv besser als ein geordnetes. Daher ist einen jährlichen Schnappschuss der Vereinsdaten zu archivieren sehr viel wichtiger.

Bewertung: 1/8

(V2) Verfassen / (V3) Bearbeiten / (V4) Löschen: Ordnungskriterien sind eine abgeleitete Forderung des Gesetzgebers, wobei der Verein durch diese ebenfalls

profitiert. Dahingegen ist das Verändern archivierter Daten nur eine Forderung des Vereins, wobei dieser durch das Verändern weniger profitieren würde, als von Ordnungskriterien. Deswegen sind diese viel wichtiger.

Bewertung: 6

A.1.4 (A6) Indexierung

(A7) Unabhängigkeit: Die eindeutige Indexierung archivierter Datensätze ist wichtiger, weil der Fall, dass die Unabhängigkeit archivierter Daten von ihrem Erzeugersystem zu tragen kommt, erst in einigen Jahren eintreten kann, aber nicht muss. Im Gegensatz dazu profitiert der Verein sowie der Gesetzgeber bei jedem Zugriff auf archivierte Daten davon, dass einzelne dieser Daten eindeutig identifiziert werden können. Deswegen ist die Indexierung viel wichtiger.

Bewertung: 4

(A8) Agilität: Agilität ist wichtiger, weil sich schon kleine Änderungen organisatorischer Strukturen des Vereins auf dessen Datenbankschema auswirken und somit bei fehlender Agilität des Archivs zum Verlust von Daten führen kann. Dahingegen sind Metadaten zum Verwalten der archivierten Daten auch wichtig, jedoch nicht so wichtig wie das Verhindern von Verlusten.

Bewertung: 1/3

(V1) Jährl. Schnappsch.: Metadaten sind als etwas wichtiger zu bewerten, weil beide Anforderungen zwar das effiziente Wiederauffinden archivierter Daten ermöglichen, jedoch Metadaten auch die Nachhaltigkeit dieser Daten unterstützen.

Bewertung: 1/2

(A9) Metadaten: Metadaten sind als etwas wichtiger zu bewerten, weil beide Anforderungen zwar das effiziente Wiederauffinden archivierter Daten ermöglichen, jedoch Metadaten auch die Nachhaltigkeit dieser Daten unterstützen.

Bewertung: 1/2

(A11) Zugriffsschutz: Dass Datensätze nicht mehr eindeutig identifizierbar sind, würden den Verein sowie den Gesetzgeber bei der Arbeit mit dem Archiv des VGBS dauerhaft behindern. Dahingegen, wenn sensible Daten in Folge des Fehlens eines Zugriffsschutzes in die falschen Hände gelangen kann dies Folgen für den Verein sowie für dessen Kunden haben. Aus diesem Grund ist der Zugriffsschutz viel wichtiger.

Bewertung: 1/6

(A12) Unveränderbarkeit: Die Unveränderbarkeit ist gesetzlich vorgeschrieben und es drohen dem Verein rechtliche Folgen wenn diese nicht gegeben ist. Zudem kann es schwerwiegende Konsequenzen haben für den VGBS haben, wenn Daten unnachweisbar manipuliert wurden. Vergleicht man dies, mit den Folgen eines fehlenden Index, nämlich dass Daten nur schwer identifiziert werden können, sieht man,

dass die Unveränderbarkeit viel wichtiger ist.

Bewertung: 1/6

(A13) Angemessenheit: Die Angemessenheit ist gesetzlich vorgeschrieben und es drohen dem Verein rechtliche Folgen wenn diese nicht gegeben ist. Jedoch ist nur sehr unpräzise beschrieben, was eine angemessene Zeit zum Zugriff auf archivierte Daten ist. Hinzu kommt, dass ein eindeutiger Index essentiell für die Verwaltung der archivierten Daten ist, weil er das eindeutige Identifizieren eines Datensatzes ermöglicht. Aus diesem Grund ist die Indexierung viel wichtiger.

Bewertung: 1/7

(V1) Jährl. Schnappsch.: Einen jährlichen Schnappschuss der Vereinsdaten zu besitzen, ist eine Forderung des Vereins, um der Aufbewahrungspflicht seiner Aufzeichnungen nachzukommen. Dahingegen ist die Indexierung essentiell für die Verwaltung der archivierten Daten, wovon sowohl der Verein, als auch der Gesetzgeber profitiert. Ist der Schnappschuss jedoch unvollständig und repräsentiert nicht sämtliche dem entsprechenden Jahr zugehörigen Aufzeichnungen, hätte dies rechtliche Konsequenzen für den VGBS. Wohingegen sich indexierte Daten letztendlich doch irgendwann auffinden. Deswegen ist jährliche Schnappschuss sehr viel wichtiger.

Bewertung: 1/8

(V2) Verfassen / (V3) Bearbeiten / (V4) Löschen: Die Indexierung der archivierten Daten ist essentiell für der deren effiziente Verwaltung. Dahingegen ist das Verändern von archivierten Daten zwar eine organisatorische Notwendigkeit für den Verein geworden, jedoch erfüllt dieses auch nur einen geringen Zweck, wenn die veränderten Daten nicht ohne weiteres verwaltet werden können. Deshalb ist die Indexierung sehr viel wichtiger.

Bewertung: 8

A.1.5 (A7) Unabhängigkeit

(A8) Agilität: Prinzipiell ist die Agilität wichtiger, weil sich schon kleine organisatorische Veränderungen direkt auf die relationale Datenbank auswirken können. Solche Veränderungen sind relativ wahrscheinlich, im Gegensatz dazu, dass das Teile oder das gesamte Erzeugersystem der archivierten Daten ausgetauscht werden müssen. Daher ist die Agilität wichtiger.

Bewertung: 1/4

(A9) Metadaten: Metadaten sind wichtiger, weil mit deren Hilfe archivierte Daten über deren gesamte Aufbewahrungsfrist effizient gesucht und identifiziert werden können. Zudem unterstützen sie die Nachhaltigkeit dieser Daten. Im Gegensatz dazu unterstützt die Unabhängigkeit archivierter Daten von ihrem Erzeugersystem zwar ebenfalls die Nachhaltigkeit dieser Daten und sogar noch besser als Metadaten, jedoch dürfte es selten der Fall sein, dass archivierte Daten durch das obsolet werden von Hardware bzw. Software verloren gehen. Wohingegen der effiziente Zugriff auf

diese Daten immer gefragt ist.

Bewertung: 1/4

(A10) Zugriffsschutz: Zugriffsschutz ist viel wichtiger, weil dieser dauerhaft gewährleistet sein muss, damit die sensiblen Daten des Vereins nicht in falsche Hände fallen. Im Gegensatz dazu ist die Unabhängigkeit archivierter Daten nur gefragt, wenn Teile oder das gesamte Erzeugersystem ausfällt bzw. obsolet wird. Dies kann jedoch erst in einigen Jahren eintreffen und selbst dann lassen sich Hardware-Komponenten austauschen bzw. Software-Umgebungen emulieren.

Bewertung: 1/7

(A11) Unveränderbarkeit: Die Unveränderbarkeit ist viel wichtiger, weil sie zum einen gesetzlich vorgeschrieben ist und zum anderen über die gesamte Aufbewahrungsdauer archivierter Daten relevant ist, damit die Authentizität dieser sichergestellt wird. Im Gegensatz dazu ist die Unabhängigkeit archivierter Daten nur gefragt, wenn Teile oder das gesamte Erzeugersystem ausfällt bzw. obsolet wird. Dies kann jedoch erst in einigen Jahren eintreffen und selbst dann lassen sich Hardware-Komponenten austauschen bzw. Software-Umgebungen emulieren.

Bewertung: 1/7

(A12) Angemessenheit: Zum einen ist die Angemessenheit gesetzlich vorgeschrieben und zum anderen möchte der Verein auch nicht jedes mal lange Zeit warten müssen um kleine Änderungen an seinen archivierten Daten vorzunehmen. Im Gegensatz dazu ist die Unabhängigkeit archivierter Daten nur gefragt, wenn Teile oder das gesamte Erzeugersystem ausfällt bzw. obsolet wird. Dies kann jedoch erst in einigen Jahren eintreffen und selbst dann lassen sich Hardware-Komponenten austauschen bzw. Software-Umgebungen emulieren. Kann dies jedoch nicht geschehen, so kann es im schlimmsten Fall dazu kommen, dass sämtliche archivierten Daten verloren gehen. Aus diesem Grund ist die Unabhängigkeit wichtiger

Bewertung: 4

(V1) Jährl. Schnappsch.: Einen jährlichen Schnappschuss seiner Daten zu besitzen ist dem Verein viel wichtiger. Der Hintergrund dessen ist, dass der Verein diesen braucht um der rechtlichen Aufbewahrungsfrist seiner Daten nachzukommen, was der Hauptgrund für die Archivierung der Vereinsdaten ist. Im Gegensatz dazu ist die Unabhängigkeit archivierter Daten nur gefragt, wenn Teile oder das gesamte Erzeugersystem ausfällt bzw. obsolet wird. Dies kann jedoch erst in einigen Jahren eintreffen und selbst dann lassen sich Hardware-Komponenten austauschen bzw. Software-Umgebungen emulieren.

Bewertung: 1/7

(V2) Verfassen / (V3) Bearbeiten / (V4) Löschen: Das Verändern archivierter Daten ist eine konkrete Anforderung des VGBS und dadurch prinzipiell in dessen Augen wichtig. Da der Verein dies jedoch durchführt, damit diese Daten, im

Falle der Auswertung im Rahmen eines Gerichtsprozess oder einer Außenprüfung, die reellen Gegebenheiten darstellen, wäre es wohl fataler, wenn diese Daten auf Grund fehlender Unabhängigkeit der archivierten Daten von ihrem Erzeugersystem noch bevor solch einer rechtlich angeordneten Auswertung verloren gingen. Deswegen ist die Unabhängigkeit der archivierten Daten wichtiger.

Bewertung: 5

A.1.6 (A8) Agilität

(A9) Metadaten: Agilität ist wichtiger, weil sich schon kleine Änderungen organisatorischer Strukturen des Vereins auf dessen Datenbankschema auswirken und somit bei fehlender Agilität des Archivs zum Verlust von Daten führen kann. Dahingegen sind Metadaten zum Verwalten der archivierten Daten auch wichtig, jedoch nicht so wichtig wie as Verhindern von Verlusten.

Bewertung: 4

(A10) Zugriffsschutz: Der Zugriffsschutz ist viel wichtiger, weil es ungewiss ist, ob sich die organisatorischen Strukturen jemals ändern, wohingegen der Zugriffsschutz unabdinglich ist, damit die sensiblen Vereinsdaten nicht in falsche Hände fallen. Wäre dieser nicht gegeben, so hätte dies unter Umständen unmittelbare Folgen für den VGBS und seine Kundschaft. Dahingegen beschränken sich die unmittelbaren Folgen fehlender Agilität, also dass Daten verloren gehen, hauptsächlich auf den Verein.

Bewertung: 1/7

(A11) Unveränderbarkeit: Die Unveränderbarkeit der archivierten Daten ist gesetzlich vorgeschrieben. Jedoch wäre es wohl fataler für den Gesetzgeber, wenn auf Grund von Veränderungen organisatorischer Strukturen des Vereins sich Daten gar nicht wiederherstellen lassen. Grundlegend ist dies aber nicht so schlimm, als wenn Daten nicht nachweisbar manipuliert werden können woraufhin der Verein oder ein Kunde des Vereins zu Schaden kommt, wie es in [Abschnitt 4.1](#) der Fall ist. Aus diesem Grund ist die Unveränderbarkeit wichtiger.

Bewertung: 1/4

(A12) Angemessenheit: Ein Archiv das etwas länger braucht um seine gespeicherten Daten bereitzustellen ist nicht so verheerend, wie eins, dass seine Daten nach einer Änderung organisatorischer Strukturen nicht wiederherstellen kann. Braucht das Archiv jedoch jedes mal eine sehr lange Zeit um seine Daten wiederherzustellen, wäre dies auch eine ständige Belastung für den Verein, wohingegen die eben erwähnten Veränderungen vielleicht gar nicht erst eintreffen können. Da diese Möglichkeit jedoch prinzipiell besteht ist die Agilität des Archivs wichtiger.

Bewertung: 3

(V1) Jährl. Schnappsch.: Der jährliche Schnappschuss ist eine Anforderung des VGBS, welche dieser stellt um den rechtlichen Aufbewahrungsfristen seiner Daten

gerecht zu werden. Im diesem Kontext ist jedoch auch die Agilität eines Archivs wichtig, weil den Aufbewahrungspflichten nicht nachgekommen werden kann, wenn sich archivierte Daten auf Grund struktureller Änderungen des Archivs nicht wieder herstellen lassen. Da es jedoch ungewiss ist, ob es jemals zu Veränderungen in der Organisation des Vereins kommt ist der jährliche Schnappschuss, als wichtiger zu bewerten.

Bewertung: 1/5

(V2) Verfassen / (V3) Bearbeiten / (V4) Löschen: Das Verändern archivierter Daten ist eine konkrete Anforderung des VGBS und dadurch prinzipiell in dessen Augen wichtig. Da der Verein dies jedoch durchführt, damit diese Daten, im Falle der Auswertung im Rahmen eines Gerichtsprozess oder einer Außenprüfung, die reellen Gegebenheiten darstellen, wäre es wohl fataler, wenn diese Daten auf Grund fehlender Agilität des Archivs noch bevor solch einer rechtlich angeordneten Auswertung verloren gingen. Deswegen ist die Agilität des Archivs daher wichtiger.

Bewertung: 5

A.1.7 (A9) Metadaten

(A10) Zugriffsschutz: Ein funktionierender Zugriffsschutz besonders wichtig, um die archivierten Daten vor unberechtigten Personen zu schützen. Wäre dieser nicht gegeben und sensible Daten fallen in die Hände destruktiver Personen, so hätte dies gravierende unmittelbare Folgen für den Verein und dessen Kunden. Dagegen profitiert der VGBS sowie der Gesetzgeber zwar auch vom effizienten verwalten archivierter Daten, wie es mit Metadaten möglich ist, jedoch sind nur schwer verwaltbare Daten besser, als das oben beschriebene Szenario. Deswegen ist der Zugriffsschutz viel wichtiger.

Bewertung: 1/7

(A11) Unveränderbarkeit: Die Unveränderbarkeit ist besonders wichtig, um die Authentizität der archivierten Daten zu schützen. Wären diese nicht unveränderbar, so könnten nicht nachweisbare Manipulationen an diesen Daten vorgenommen werden, sodass diese dem VGBS und seine Kunden schaden können (siehe [Abschnitt 4.1](#)). Über dies ist die Unveränderbarkeit gesetzlich vorgeschrieben. Dagegen profitiert der VGBS sowie der Gesetzgeber zwar auch vom effizienten verwalten archivierter Daten, wie es mit Metadaten möglich ist, jedoch sind nur schwer verwaltbare Daten besser, als das oben beschriebene Szenario. Deswegen ist die Unveränderbarkeit archivierter Daten viel wichtiger.

Bewertung: 1/8

(A12) Angemessenheit: Metadaten sind viel wichtiger, weil die gesetzlich vorgeschriebene Angemessenheit von von Metadaten direkt profitiert. Zusätzlich dazu unterstützen Metadaten die Nachhaltigkeit der archivierten Daten.

Bewertung: 7

(V1) Jährl. Schnappsch.: Einen jährlichen Schnappschuss der Vereinsdaten zu archivieren ist viel wichtiger, weil damit im Falle einer Gerichtsverhandlung bzw. steuerrechtlichen Außenprüfung die Nachweisbarkeit der Aufzeichnungen des VGBS sichergestellt wird. Dahingegen dienen Metadaten unter anderem auch dem Verwalten dieser Aufzeichnungen, sodass es wichtiger ist sämtliche Aufzeichnungen zu besitzen, diese jedoch schlecht verwaltbar sind, als andersherum.

Bewertung: 1/7

(V2) Verfassen / (V3) Bearbeiten / (V4) Löschen: Metadaten sind viel wichtiger, weil mit ihnen die archivierten Daten effizient verwaltet werden können. Zusätzlich dazu unterstützen sie die Nachhaltigkeit dieser Daten. Von diesen Vorteilen profitiert sowohl der VGBS, als auch der Gesetzgeber, wenn diesem die Daten zu Verfügung gestellt werden. Dahingegen profitiert vom Verfassen archivierter Daten in erster Linie nur der Verein.

Bewertung: 7

A.1.8 (A10) Zugriffsschutz

(A11) Unveränderbarkeit: Unveränderbarkeit ist etwas wichtiger, weil diese gesetzlich vorgeschrieben ist. Dahingegen gleichen sich die Folgen, wenn eine der beiden Anforderungen nicht erfüllt wird im Hinblick darauf, dass der Verein bzw. seine Kunden Schaden nehmen könnten, wobei in diesem Kontext noch eine Strafe wegen der fehlenden Unveränderbarkeit der archivierten Daten hinzukommt.

Bewertung: 1/2

(A12) Angemessenheit: Ein funktionsfähiger Zugriffsschutz ist viel wichtiger, weil dieser sicherstellt, dass die sensiblen Vereinsdaten nicht in falsche Hände gelangen. Dies könnte ansonsten verheerende unmittelbare Folgen für den Verein bzw. dessen Kundschaft nach sich ziehen. Im Gegensatz dazu ist die Angemessenheit zwar gesetzlich vorgeschrieben, jedoch nicht sehr eindeutig formuliert. Zudem werden der Gesetzgeber bzw. VGBS auch länger auf den Zugriff auf archivierte Daten warten können, wenn diese dafür im Vorfeld vor unberechtigten Zugriffen geschützt wurden.

Bewertung: 6

(V1) Jährl. Schnappsch.: Der jährliche Schnappschuss ist viel wichtiger, weil mit diesem die finanzielle Lage des Vereins der vergangenen Jahre nachgewiesen werden kann, was eines der Hauptziele der Archivierung des Vereins darstellt. Dahingegen ist der Zugriffsschutz nur wichtig, wenn dieses Ziel zumindest teilweise erreicht ist.

Bewertung: 1/6

(V2) Verfassen / (V3) Bearbeiten / (V4) Löschen: Ein funktionsfähiger Zugriffsschutz ist extrem viel wichtiger, weil dieser sicherstellt, dass die sensiblen Vereinsdaten nicht in falsche Hände gelangen. Dies könnte ansonsten verheerende

unmittelbare Folgen für den Verein bzw. dessen Kundschaft nach sich ziehen. Dahingegen ist das Verändern archivierter Daten ein zwar eine organisatorische Notwendigkeit für den Verein geworden, jedoch wird dieser auch darauf verzichten können, wenn dafür seine Daten vor unberechtigten Zugriffen

Bewertung: 9

A.1.9 (A11) Unveränderbarkeit

(A13) Angemessenheit: Die Unveränderbarkeit der archivierten Daten ist viel wichtiger, weil mit dieser u.a. unberechtigte und destruktive Manipulationen aufgedeckt werden können (siehe [Abschnitt 4.1](#)). Dahingegen ist die Angemessenheit nicht sehr präzise formuliert und lässt offen, bei welcher Zeitspanne es sich um eine angemessene Zeit handelt.

Bewertung: 7

(V1) Jährl. Schnappsch.: Der jährliche Schnappschuss der Vereinsdaten ist viel wichtiger, weil dieser die Nachweisbarkeit der jährlichen Aufzeichnungen des VGBS sicherstellt. Dahingegen dient die Unveränderbarkeit in erster Linie dem Schutz dieser Aufzeichnungen.

Bewertung: 1/7

(V2) Verfassen / (V3) Bearbeiten / (V4) Löschen: Die Unveränderbarkeit der archivierten Daten ist extrem viel wichtiger, weil mit dieser u.a. unberechtigte und destruktive Manipulationen aufgedeckt werden können (siehe [Abschnitt 4.1](#)). Über dies kommt hinzu, dass die Unveränderbarkeit gesetzlich vorgeschrieben ist und dem VGBS rechtliche Strafen drohen, wenn diese nicht eingehalten wird. Dahingegen ist das Verfassen archivierter Daten zwar eine organisatorische Notwendigkeit für den Verein geworden, jedoch wird dieser auch darauf verzichten können, wenn dafür eben erwähnten Folgen umgangen werden könnten.

Bewertung: 9

A.1.10 (A12) Angemessenheit

(V1) Jährl. Schnappsch.: Der jährliche Schnappschuss der Vereinsdaten ist viel wichtiger, weil dieser die Nachweisbarkeit der jährlichen Aufzeichnungen des VGBS sicherstellt. Dahingegen ist die Angemessenheit zwar gesetzlich vorgeschrieben, jedoch zum einen unpräzise formuliert, sodass offen ist was genau eine angemessene Zeit zum Wiedererlangen archivierter Daten ist und zum anderen sind die rechtlichen Strafen wohl höher, wenn archivierte Daten einerseits schnell wiedergefunden werden können, diese andererseits nicht sämtliche Aufzeichnungen des Vereins enthalten.

Bewertung: 1/7

(V2) Verfassen / (V3) Bearbeiten / (V4) Löschen: Der Verein profitiert prinzipiell von einem schnellen Zugriff auf seine archivierten Daten. Auf der anderen Seite greift dieser jedoch fast ausschließlich auf diese zu, um die archivierten Daten nachträglich anzupassen. Bietet das Archiv jedoch nur einen schnellen Zugriff und

würde dahingegen keine Anpassungen ermöglichen, würde der VGBS einen geringen Nutzen von seinem Archiv haben. Da das schnelle Bereitstellen archivierter Daten jedoch von Gesetzgeber gefordert wird und der Verein diesen auch benötigt um beispielsweise Kundenanfragen zu beantworten, ist die Angemessenheit wichtiger.

Bewertung: 4

A.1.11 (V1) Jährlicher Schnappschuss

(V2) Verfassen / (V3) Bearbeiten / (V4) Löschen: Der jährliche Schnappschuss der Vereinsdaten ist viel wichtiger, weil dieser die Nachweisbarkeit der jährlichen Aufzeichnungen des VGBS sicherstellt. Wäre dies nicht gegeben, so könnte das rechtliche Konsequenzen für den Verein nach sich ziehen. Dagegen ist das anpassen dieser nur eine Maßnahme um diese Aufzeichnungen nachträglich anzupassen, wenn sich im aktuellen Jahr Gegebenheiten ereignen die für den VGBS logisch ins vorhergehende gehören. Dies ist jedoch vom Gesetzgeber nicht so vorgesehen bzw. unter Umständen nicht gewollt.

Bewertung: 7

A.1.12 (V2) Verfassen

(V3) Bearbeiten: Das Verfassen und Bearbeiten archivierter Daten ist von gleicher Wichtigkeit für den VGBS. Der Hintergrund dessen ist, dass sich Anpassungen von archivierten Daten häufig nur auf gezahlte Beiträge beziehen. Hierbei ist es jedoch offen, ob der gesamte Beitrag eines Monats nachgezahlt wurde und somit ein komplett neuer Datensatz verfasst wird oder nur noch der Rest des monatlichen Beitrages nachgezahlt wurde, sodass der Datensatz des entsprechenden Monats bearbeitet werden muss.

Bewertung: 1

(V4) Löschen: Das Verfassen eines archivierten Datensatzes ist wichtiger, weil sich Anpassungen von archivierten Daten häufig nur auf gezahlte Beiträge beziehen, wobei diese in der Regel nur nachgezahlt werden und nur selten nachträglich gelöscht.

Bewertung: 4

A.1.13 (V3) Bearbeiten

(V4) Löschen: Das Bearbeiten eines archivierten Datensatzes ist wichtiger, weil sich Anpassungen von archivierten Daten häufig nur auf gezahlte Beiträge beziehen, wobei diese in der Regel nur nachgezahlt werden und nur selten nachträglich gelöscht.

Bewertung: 4

A.2 Funktionsweise der Archivierungsmethoden

In diesem Abschnitt werde ich die detaillierte Funktionsweise der datenbankinternen und datenbankexternen Methode vorstellen.

A.2.1 Datenbankinterne Methode

Die Funktionsweise der datenbankinternen Methode werde ich an der Mitglied-Tabelle meines im Vorfeld eingeführten Beispiels illustrieren (siehe [Abschnitt 4.1](#) „Begleitendes Beispiel“). Dabei werde ich erläutern wie das Einfügen, Bearbeiten und Löschen bzw. Archivieren eines Datensatzes in diesem Kontext funktioniert.

Einfügen: Beim Einfügen eines neuen Datensatzes ist der Gültigkeitsbeginn dessen davon Abhängig, welches Verwaltungsjahr in der Anwendung vor dem Einfügen vom Nutzer spezifiziert wurde. Dabei besitzt jeder neu hinzugefügte Datensatz zunächst ein unbegrenzt Gültigkeitsende. Somit ist jeder dieser Datensätze vom Verwaltungsjahr in dem er hinzugefügt wurde bis zunächst unbegrenzte Zeit in der Anwendung operativ verfügbar. Dies bedeutet, dass egal in welchem Verwaltungsjahr ein Datensatz neu hinzugefügt wurde, er auch im operativen Verwaltungsjahr dem Nutzer von der Anwendung bereitgestellt wird. Dahingegen ist dieser, wie eben besprochen, in einem Verwaltungsjahr vor seinem Einfügen für den Nutzer innerhalb der Anwendung nicht vorhanden. Weiterhin erhält jeder neue Datensatz einen aktuellen Transaktionszeitstempel.

Ich möchte dies an meinem Beispiel illustrieren (siehe [Abbildung A.1](#)): Der Datensatz Danny wird zusammen mit seiner Zuweisung zur Sportgruppe „SGKF“ in das Verwaltungsjahr **2015** aufgenommen und erhält somit als Gültigkeitsbeginn dieses Jahr sowie einen aktuellen Transaktionszeitstempel. Zum Zeitpunkt **25.05.15 - 12:49:34** als Tim diesen in die Datenbank aufgenommen hat, handelte es sich hierbei um das operative Verwaltungsjahr. Beim Wechsel in das Jahr 2016, wechselt auch das operative Verwaltungsjahr zu diesem Jahr. Sowohl im vergangenen Verwaltungsjahr 2015, als auch im neuen Verwaltungsjahr 2016 werden Danny Datensätze jedoch dem Nutzer von der Anwendung noch bereitgestellt, weil deren Gültigkeitsbeginn gleich 2015 ist und ihr Gültigkeitsende auf einen unbegrenzt entfernten Zeitpunkt festgelegt wurde. Dadurch können diese weiterhin für organisatorische oder administrative Zwecke genutzt werden. Betrachtet der Nutzer dahingegen das Verwaltungsjahr 2014, werden ihm diese Datensätze von der Anwendung nicht zur Verfügung gestellt und können dementsprechend dort nicht genutzt werden, weil ihr Gültigkeitsbeginn größer als dieses Verwaltungsjahr ist.

Bearbeiten: Wird ein Datensatz bearbeitet, so existieren in diesem Kontext zwei Szenarien:

1. Ein Szenario besteht darin, dass der Nutzer einen Datensatz bearbeitet dessen Gültigkeitsbeginn gleich dem spezifizierten Verwaltungsjahr ist. Hierbei wird dieser Datensatz mit einer bearbeiteten neuen Version überschrieben. Die Transaktionszeit bewirkt dabei, dass dieses Überschreiben nur logisch stattfindet. Das heißt die bearbeitete neue Version wird der Tabelle als eigenständiges Tupel mit aktuellem Transaktionszeitstempel hinzugefügt. Der Transaktionsbeginn des neuen Tupels sowie das Transaktionsende des ursprünglichen zeigen dabei auf den Zeitpunkt der Modifikation.



Abbildung A.1: Tim fügt Danny der Datenbank hinzu.

- Wird ein Datensatz in einem Verwaltungsjahr bearbeitet, das ungleich seines Gültigkeitsbeginn ist, so wird eine neue bearbeitete Version des gleichen Datensatzes der Tabelle hinzugefügt. Diese ist fortan von diesem Verwaltungsjahr an gültig. Die Gültigkeit der ursprünglichen unveränderten Version endet dahingegen in diesem Verwaltungsjahr. Damit bleiben die Daten der vorhergehenden Verwaltungsjahre erhalten, so wie der Verein dies fordert. Durch die Natur der Transaktionszeit führt in diesem Zusammenhang jede Veränderung von Gültigkeitszeiten, ebenfalls zu einer weiteren von dort an aktuellen Version des Datensatzes. Damit wird sichergestellt, dass jegliche Veränderung von Daten protokolliert wird, so wie es die Unveränderbarkeit fordert.

Im Kontext dieses Szenarios kann das Verwaltungsjahr in dem ein Datensatz bearbeitet wird nur größer als dessen Gültigkeitsbeginn sein, weil, wie bereits erwähnt, der Datensatz in älteren Verwaltungsjahren nicht von der Anwendung bereitgestellt wird und deswegen dort nicht bearbeitet werden kann.

Damit dies deutlicher wird, möchte ich beide Szenarien wieder an meinem Beispiel aus [Abschnitt 4.1](#) erklären:

Die folgenden Ausführungen zu Tims Manipulationen bezüglich Dannys Sportgruppenzuweisung können in [Abbildung A.2](#) betrachtet werden. Veränderungen der Daten werden in grau hinterlegt dargestellt. Jedes Tupel erhält einen Buchstaben von A bis D. A beschreibt dabei Dannys ursprüngliche Zuweisung zur Sportgruppe „SGKF“.

Szenario 2

Zum 01.01.16 wechselt das operative Verwaltungsjahr von 2015 zu 2016. Am **08.04.16 um 12:45:56 Uhr** vermerkt Tim, dass Danny am 01.05.16 aus der Sportgruppe

Ist in

	<u>MSgNr</u>	<u>MitglNr</u>	<u>SgNr</u>	<u>Beitritt</u>	<u>Austritt</u>	<u>GültigSeit</u>	<u>GültigBis</u>	<u>ErfasstAm</u>	<u>GelöschtAm</u>
A	1	1	1	01.07.15	-	2015	2143	2015-05-01 12:49:34	9999-12-31 23:59:59

↓ 2016-04-08 12:45:56

Ist in

	<u>MSgNr</u>	<u>MitglNr</u>	<u>SgNr</u>	<u>Beitritt</u>	<u>Austritt</u>	<u>GültigSeit</u>	<u>GültigBis</u>	<u>ErfasstAm</u>	<u>GelöschtAm</u>
A	1	1	1	01.07.15	-	2015	2143	2015-05-01 12:49:34	2016-04-08 12:45:56
B	1	1	1	01.07.15	-	2015	2016	2016-04-08 12:45:56	9999-12-31 23:59:59
C	1	1	2	01.07.15	01.05.16	2016	2143	2016-04-08 12:45:56	9999-12-31 23:59:59

↓ 2016-04-08 12:47:12

Ist in

	<u>MSgNr</u>	<u>MitglNr</u>	<u>SgNr</u>	<u>Beitritt</u>	<u>Austritt</u>	<u>GültigSeit</u>	<u>GültigBis</u>	<u>ErfasstAm</u>	<u>GelöschtAm</u>
A	1	1	1	01.07.15	-	2015	2143	2015-05-01 12:49:34	2016-04-08 12:45:56
B	1	1	1	01.07.15	-	2015	2016	2016-04-08 12:45:56	2016-04-08 12:47:12
D	1	1	2	01.07.15	-	2015	2016	2016-04-08 12:47:12	9999-12-31 23:59:59
C	1	1	2	01.07.15	01.05.16	2016	2143	2016-04-08 12:45:56	9999-12-31 23:59:59

Abbildung A.2: Ablauf der Geschehnisse

„SGKF“ austreten wird. Parallel dazu trägt er Danny für den Zeitraum vom 01.07.15 bis 01.05.16 in die Sportgruppe „Freizeitsport“ ein. Dies bedeutet, dass der mit dem Buchstaben A gekennzeichnete Datensatz im Verwaltungsjahr 2016 bearbeitet wird. Also handelt es sich hierbei um Szenario 2.

Da der Gültigkeitsbeginn von A 2015 ist, wird eine neue Version in die Tabelle eingefügt, welche ab dem Verwaltungsjahr 2016 gültig ist. Dafür wird zunächst das Gültigkeitsende von Datensatz A auf 2016 festgelegt. Weil es sich hierbei um eine Modifikationen handelt, bewirkt die Transaktionszeit, dass ein neuer Datensatz der Tabelle hinzugefügt wird. In diesem Fall ist dies Datensatz B. Dieser beinhaltet die Modifikation und erhält einen aktuellen Transaktionszeitstempel. Parallel dazu wird das Transaktionsende von A auf den Zeitpunkt der Modifikation festgelegt. Damit ist A nicht mehr aktuell.

Im Anschluss daran wird Datensatz C hinzugefügt, welcher Dannys Zuweisung zur Sportgruppe „Freizeitsport“ für das Verwaltungsjahr 2016 beinhaltet. Diese behält das Gültigkeitsende der ursprünglichen Version A, also 2143 und ist fortan ab dem Verwaltungsjahr 2016 gültig. Damit diese Veränderung protokolliert wird, erhält C ebenfalls einen aktuellen Transaktionszeitstempel.

Szenario 1

Am **08.04.16 um 12:47:12 Uhr** manipuliert Tim Dannys archivierte Daten, damit auch diese so aussehen, als wären seine zu hoch gezahlte Beiträge gerechtfertigt. Hierfür wechselt Tim mit Hilfe der Verwaltungsanwendung in das Verwaltungsjahr 2015. Dort wird ihm der in [Abbildung A.2](#) dargestellte Datensatz B bereitgestellt. Nun bearbeitet Tim diesen und ändert dessen Sportgruppenreferenz auf „Freizeitsport“. Damit handelt es sich hierbei, um Szenario 1. Daraus folgt, dass Version D

der Tabelle hinzugefügt wird. Diese erhält einen aktuellen Transaktionszeitstempel. Die ursprüngliche Version B ist noch in der Tabelle enthalten und erhält als Transaktionsende den Zeitpunkt der Modifikation 08.04.16 12:47:12. Da sowohl Version B, als auch C den gleichen Gültigkeitszeitstempel aufweisen, überschreibt C die Version B im Verwaltungsjahr 2015.

Löschen: Soll ein Datensatz in einem Verwaltungsjahr gelöscht werden, so existieren hierfür zwei Szenarien. In beiden Szenarien werden ebenfalls jene Datensätze gelöscht die mit dem zu Löschen in einer kaskadierenden Beziehung stehen und im gleichen Zeitraum gültig sind. Damit bleibt die referentielle Integrität im temporalen Sinn erhalten (siehe [Abschnitt 2.2.2.3](#) „Integritätsbedingungen“).

1. Der zu löschende Datensatz ist operativ. In diesem Fall wird sein Gültigkeitsende auf das Verwaltungsjahr festgelegt in dem dieser gelöscht wird. Die Transaktionszeit bewirkt hierbei wieder, dass ein neuer Datensatz der Tabelle hinzugefügt wird, welcher diese Änderung enthält. Somit wurde dieser gleichzeitig archiviert.
2. Der zu löschende Datensatz ist bereits archiviert, so wird sein Gültigkeitsende ebenfalls auf das Verwaltungsjahr des Löschens festgesetzt. Die Transaktionszeit bewirkt hierbei wieder, dass ein neuer Datensatz der Tabelle hinzugefügt wird, welcher diese Änderung enthält. Hierbei werden jedoch auch sämtliche Ausprägungen dieses Datensatzes deren Gültigkeitsbeginn nach diesem Verwaltungsjahr liegt aus der Datenbank logisch gelöscht. Dies bedeutet, dass das Transaktionsende dieser Datensätze auf den Zeitpunkt des Löschens festgelegt wird.

Hierbei geht die datenbankinterne Methode davon aus, dass die zeitlich nachfolgenden Ausprägungen eines Datensatzes, sowie auf diese referenzierende Datensätze nicht entstanden wären, wenn es die zu löschende Ausprägung nicht gäbe.

Zur besseren Verbildlichung dieser zwei Szenarien, werde ich wieder mein Beispiel aus [Abschnitt 4.1](#) heranziehen.

Szenario 1

Szenario 1 kann in [Abbildung A.3](#) betrachtet werden. Hierbei wurde Dannys Datensatz am **22.11.20 um 15:23:26 Uhr** im Verwaltungsjahr **2020** gelöscht. Daraufhin wird dessen Gültigkeitsende auf 2020 gesetzt. Da es sich dabei um eine Datenmodifikation handelt entsteht der mit A gekennzeichnete Datensatz. Da Dannys Sportgruppenzuweisung in einer kaskadierenden Beziehung zu ihm steht und im gleichen Zeitraum Gültig ist, wird diese ebenfalls gelöscht. Weil es sich hierbei wiederum um einen operativen Datensatz handelt, so wird mit diesem genauso verfahren.



Abbildung A.3: Löschen: Szenario 1

Szenario 2

Szenario 2 ist in [Abbildung A.4](#) illustriert. Hierbei ist anzunehmen, dass Dannys Sportgruppenzuweisung zu „SGKF“ am **22.11.20 um 15:23:16 Uhr** im Verwaltungsjahr **2020** überschrieben wurde. Dabei wurde ihm die Sportgruppe „Freizeisport“ sowie der Beitrag A zugewiesen. Nun wird Danny am 06.06.21 um 13:23:16 Uhr im Verwaltungsjahr 2017 gelöscht. Im Verwaltungsjahr 2017 ist Datensatz A gültig. Aus diesem Grund wird dessen Gültigkeitsende auf 2017 gesetzt. Da es sich hierbei um eine Datenmodifikation handelt entsteht Datensatz E. Weil Dannys Version B noch nach A gültig gewesen wäre, wird das Transaktionsende von B auf den Löszeitpunkt gesetzt. Nun werden die Datensätze gesucht, die mit Danny in einer kaskadierenden Beziehung stehen und im gleichen Zeitraum gültig sind wie Datensatz A. Dies ist Datensatz C, wodurch dieser ebenfalls gelöscht wird. Da dieser ebenfalls bereits archiviert ist, wird mit ihm genauso verfahren wie mit Datensatz A.

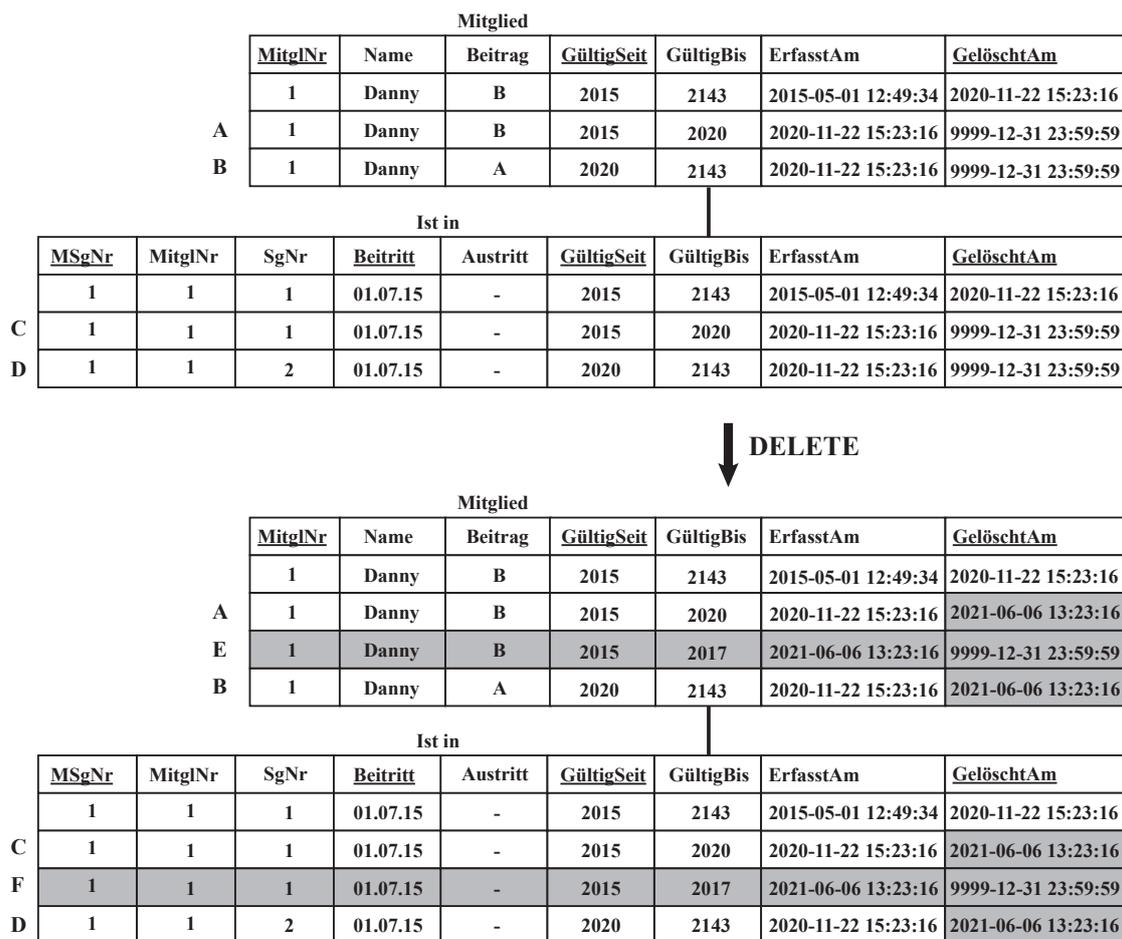


Abbildung A.4: Löschen: Szenario 2

A.2.2 Datenbankexterne Methode

In diesem Abschnitt erläutere ich die Funktionsweise der datenbankexternen Methode. Dafür werde ich zunächst den Ablauf dieser Methode erläutern und im Anschluss beschreiben wie der Zugriff auf die archivierten Daten erfolgt. Nachfolgend führe ich dann aus wie das Modifizieren archivierter Daten funktioniert und was die Auswirkungen davon sind.

Wie Eingangs beschrieben werden im Kontext der datenbankexternen Methode sämtliche Daten verändernde SQL-Anweisungen protokolliert und in einem entsprechenden Verwaltungsjahr gespeichert. Dies erfüllt im wesentlichen folgende Zwecke:

Zum einen kann mit Hilfe dieser Protokollierung von Veränderungen die gesetzlich geforderte Unveränderbarkeit sichergestellt werden. Dies ist jedoch nur in Kombination mit der Modifikationszeit möglich. Dadurch wird ersichtlich wann, welche Modifikation stattfand und jede dieser Modifikationen können im einzelnen nachverfolgt werden. Zum anderen werden mit Hilfe der SQL-Anweisungen frühere Zustände der Datenbank archiviert. Verbunden mit den Verwaltungsjahren können somit gezielt Zustände der Vereinsdaten aus vorhergehenden Jahren wiederhergestellt und modifiziert werden.

Dass SQL-Anweisungen zum Protokollieren der Modifikationen genutzt werden hat den Vorteil, dass zum Protokollieren sowie zur Wiederherstellung archivierter Daten keine besondere Anwendungslogik erforderlich ist. Bei einer Modifikation können die entsprechenden SQL-Anweisungen direkt in die Dateien gespeichert werden. Ausschließlich das Einsetzen von Parametern durch die Anwendung wird hierbei unter Umständen erforderlich. Im Falle einer Wiederherstellung archivierter Daten können diese Dateien direkt an das DBMS übermittelt werden, wobei dieses die weiteren Schritte übernimmt.

Ferner werden nur jene SQL-Befehle gespeichert, welche Veränderungen an den Daten hervorrufen. Dies umfasst somit jegliche INSERT-, UPDATE- und DELETE-Befehle. SQL-Befehle, die Änderungen am Datenbankschema verursachen oder jene zum Erstellen von Triggern bzw. Prozeduren werden ausgeschlossen, weil diese in der Verwaltungsanwendung des VGBS nicht spezifiziert werden können.

Ablauf

Im Rahmen der datenbankexternen Methode wird zunächst zum Zeitpunkt t_0 , welcher den Zeitpunkt der Inbetriebnahme der Datenbank markiert, ein initialer Schnappschuss der Datenbank angefertigt. Zu diesem Zeitpunkt ist das Datenbankschema bereits vollständig modelliert, jedoch befinden sich noch keine Daten in der Datenbank. Dieser Schnappschuss enthält somit die gesamte Schemadefinition, in Form von Tabellennamen und Constraints. Erforderlich ist dieses Vorgehen, um zu einem späterem Zeitpunkt t_n mit $n > 0$, wieder zum Datenbankzustand von t_0 gelangen zu können. Darauf aufbauend können dann die protokollierten SQL-Anweisungen wiederhergestellt werden.

Ab dem Zeitpunkt t_1 wird nun jede Daten verändernde SQL-Anweisung, die von der Anwendung an das DBMS geschickt wird, mit einem Modifikationszeitstempel versehen und in eine entsprechende Datei geschrieben, welche ein Verwaltungsjahr repräsentiert. In einer Datei werden die SQL-Anweisungen jeweils untereinander in der Reihenfolge ihrer Entstehung festgehalten. Das bedeutet, dass die erste Anweisung den ältesten Modifikationszeitstempel besitzt und die letzte den jüngsten. In der Regel werden sämtliche SQL-Anweisungen im operativen Verwaltungsjahr protokolliert. Dieses endet am 01.01. des darauffolgenden Jahres, welches ab diesem Zeitpunkt operativ ist.

Zugriff

Möchte der Nutzer dahingegen ein vorangegangenes Verwaltungsjahr betrachten und daran Modifikationen durchführen, so spezifiziert dieser in der Anwendung das gewünschte Verwaltungsjahr. Anhand dieser Spezifikation wird die entsprechende Datei von der Anwendung ausgewählt und wiederhergestellt. Hierfür muss als erstes der initiale Schnappschuss wiederhergestellt werden. Auf diesen Aufbauend werden sämtliche Verwaltungsjahre die älter sind als das spezifizierte bis zu diesem wiederhergestellt. Dies ist notwendig, weil jedes Verwaltungsjahr Modifikationen in Form von UPDATE- und DELETE-Anweisungen enthalten kann, die ohne das vorhergehende nicht möglich wären.

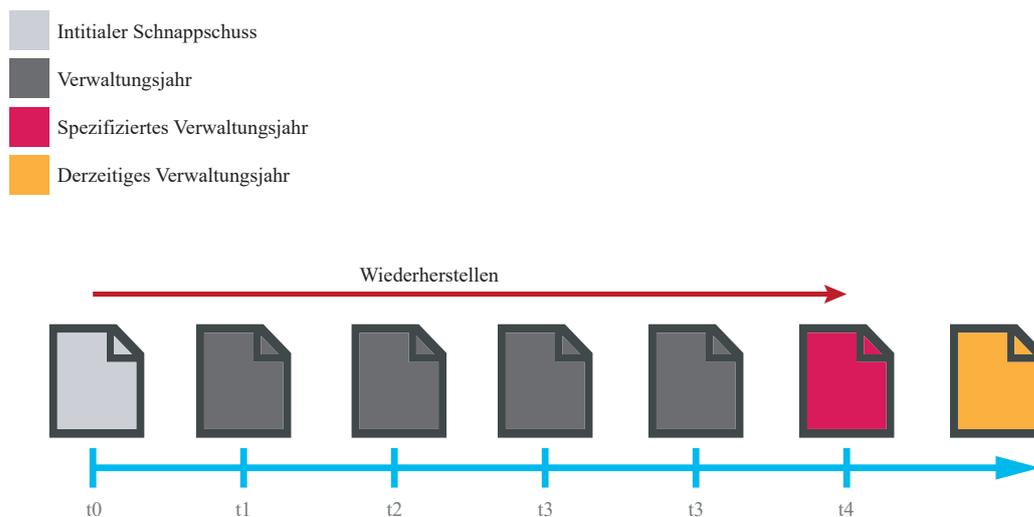


Abbildung A.5: Wiederherstellen eines vorhergehenden Verwaltungsjahres

Befindet sich der Nutzer jedoch bereits in einem archiviertes Verwaltungsjahr und möchte ein jüngeres betrachten, so ist es nur noch erforderlich vom derzeit befindlichen bis zum gewünschten Verwaltungsjahr alle Verwaltungsjahre wiederherzustellen.

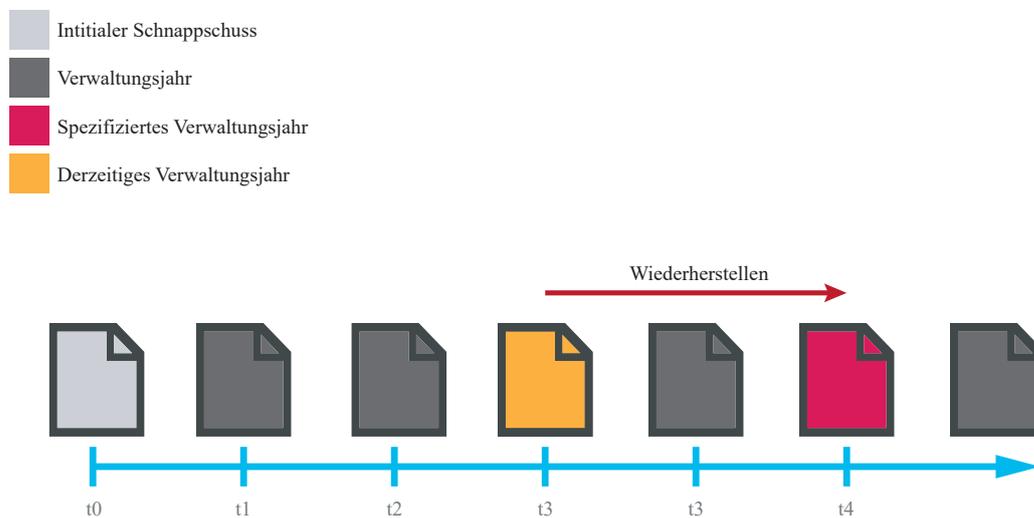


Abbildung A.6: Wiederherstellen eines jüngeren Verwaltungsjahres

Das Wiederherstellen eines Verwaltungsjahres bedeutet im einzelnen, dass die in der Datei enthaltenen SQL-Anweisungen sukzessiv in der Reihenfolge ihrer Speicherung durch das DBMS ausgeführt werden. Das heißt, dass zunächst die älteste, darauf folgend die nächst jüngere und zum Schluss die jüngste SQL-Anweisung ausgeführt wird.

Modifizieren

Nun betrachtet der Nutzer das Ausgewählte Verwaltungsjahr und kann Modifikationen an diesem vornehmen. Dabei wird jede SQL-Anweisung die Modifikationen an Daten vornimmt jeweils nur im betrachteten Verwaltungsjahr protokolliert. Damit wird sichergestellt, dass sich Veränderungen nur auf jene Daten Auswirken an denen sie vorgenommen wurden, wodurch die Veränderbarkeit der archivierten Daten im Sinne des VGBS sichergestellt wird.

Auch hierbei werden diese Anweisungen stets an das Ende der Datei angefügt und erhalten einen aktuellen Modifikationszeitstempel. Eine einmal protokollierte Modifikation wird somit unter keinen Umständen überschrieben. Dadurch wird ersichtlich wann welche Modifikation stattfand und ob diese durchgeführt wurden, als das betrachtete Verwaltungsjahr operativ war oder ob diese nachträglich vollzogen wurden. Aus diesem Grund wird hiermit die gesetzlich vorgeschriebene Unveränderbarkeit der archivierten Daten sichergestellt.

Weiterhin, da zunächst sämtliche zeitlich vorhergehenden Verwaltungsjahre wiederhergestellt werden müssen, um zu dem gewünschten zu gelangen, wirken sich Modifikationen in einem Verwaltungsjahr jeweils auch auf die zeitlich nachfolgenden Verwaltungsjahre aus. Wird Beispielsweise ein Datensatz im Verwaltungsjahr 2014 eingefügt, so ist dieser auch in 2015 und sogar im operativen Verwaltungsjahr 2016 vorhanden. Dies gilt jedoch nur, wenn dieser nicht zuvor in einem Verwaltungsjahr gelöscht wurde. In diesem Kontext ist ein gelöschter Datensatz nur in den Verwaltungsjahren verfügbar die zeitlich vor diesem liegen. Löscht ein Nutzer einen Datensatz im Verwaltungsjahr 2015, der im Verwaltungsjahr 2013 protokolliert wurde, so ist dieser nur in den Verwaltungsjahren 2013 und 2014 verfügbar, nicht aber in 2015 und darüber hinaus. Der Hintergrund dessen ist, dass die entsprechende DELETE-Anweisung zusammen mit den anderen protokollierten SQL-Anweisungen des Verwaltungsjahres 2015 ausgeführt werden, um dieses Wiederherzustellen. Somit ist der gelöschte Datensatz in 2015 nicht mehr verfügbar. Ähnlich sieht es beim Bearbeiten eines Datensatzes aus. Die modifizierte Version dessen ist aus den gleichen Gründen ebenfalls erst ab dem Verwaltungsjahr verfügbar, in dem die Modifikation stattfand.

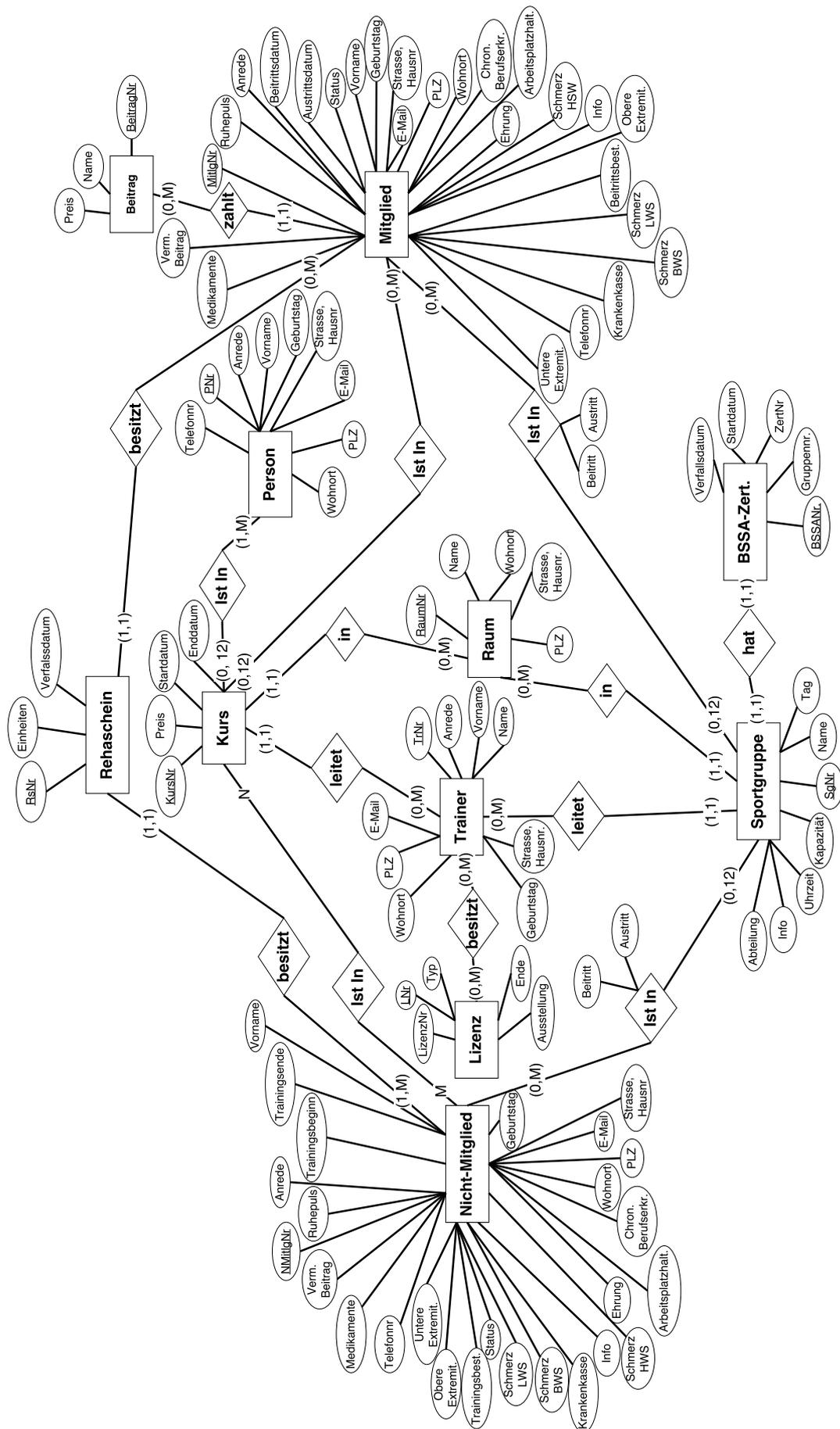


Abbildung A.7: Datenbankschema des VGBS

Literaturverzeichnis

- [BR07] Navneet Bhushan and Kanwal Rai. *Strategic decision making: applying the analytic hierarchy process*. Springer Science & Business Media, 2007. (zitiert auf Seite ix, 5 und 7)
- [dL14] Oberste Finanzbehörden der Länder. Grundsätze zur ordnungsmäßigen Führung und Aufbewahrung von Büchern, Aufzeichnungen und Unterlagen in elektronischer Form sowie zum Datenzugriff (GoBD), 2014. (zitiert auf Seite 2, 25, 27 und 29)
- [HKS95] A. Herbst, K. Küspert, and R. Schaarschmidt. *Datenarchivierungsfunktionalität in der SQL×-Norm*. Forschungsergebnisse der Fakultät für Mathematik und Informatik / Friedrich-Schiller-Universität Jena. Friedrich-Schiller-Universität, 1995. (zitiert auf Seite 25, 26, 27, 28 und 29)
- [Inn08] Swiss Federal Archives SFA Unit Innovation. Preservation. SIARD Format Description. Technical report, Technical report, Federal Department of Home Affairs, FDHA, Berne, 2008. (zitiert auf Seite 37)
- [ISO12] ISO. Space Data and Information Transfer Systems. Open Archival Information System (OAIS), Reference Model. ISO 14721:2012, International Organization for Standardization, Geneva, Switzerland, 2012. (zitiert auf Seite 25 und 29)
- [JDB⁺98] Christian S. Jensen, Curtis E. Dyreson, Michael Böhlen, James Clifford, Ramez Elmasri, Shashi K. Gadia, Fabio Grandi, Pat Hayes, Sushil Jajodia, Wolfgang Käfer, Nick Kline, Nikos Lorentzos, Yannis Mitsopoulos, Angelo Montanari, Daniel Nonen, Elisa Peressi, Barbara Pernici, John F. Roddick, Nandlal L. Sarda, Maria Rita Scalas, Arie Segev, Richard T. Snodgrass, Mike D. Soo, Abdullah Tansel, Paolo Tiberio, and Gio Wiederhold. *Temporal Databases: Research and Practice*, chapter The consensus glossary of temporal database concepts — February 1998 version, pages 367–405. Springer Berlin Heidelberg, Berlin, Heidelberg, 1998. (zitiert auf Seite 11, 12, 13, 43 und 44)
- [JLRH02] Marta H Jacinto, Giovanni R Librelotto, José C Ramalho, and Pedro R Henriques. Bidirectional conversion between xml documents and relational databases. In *The 7th International Conference on Computer Supported Cooperative Work in Design, 2002.*, pages 437–443. IEEE, 2002. (zitiert auf Seite 28 und 37)

- [Kam06a] Ulrich Kampffmeyer. Elektronische archivierung, 2006. (zitiert auf Seite 1, 25, 26 und 27)
- [Kam06b] Ulrich Kampffmeyer. Revisions sichere Archivierung im Licht neuer rechtlicher Anforderungen I-IV, 2006. (zitiert auf Seite 1, 25, 27 und 28)
- [KR97] Ulrich Kampffmeyer and Jörg Rogalla. Grundsätze der elektronischen Archivierung. *Code of Practice*, 1997. (zitiert auf Seite 25 und 29)
- [KWR98] Joachim Karlsson, Claes Wohlin, and Björn Regnell. An evaluation of methods for prioritizing software requirements. *Information and Software Technology*, 39(14):939–947, 1998. (zitiert auf Seite 5)
- [LM97] Nikos A Lorentzos and Yannis G Mitsopoulos. Sql extension for interval data. *IEEE Transactions on knowledge and Data Engineering*, 9(3):480–499, 1997. (zitiert auf Seite 14)
- [MK13] Axel C. Mühlbacher and Anika Kaczynski. Der analytic hierarchy process (ahp): Eine methode zur entscheidungsunterstützung im gesundheitswesen. *PharmacoEconomics German Research Articles*, 11(2):119–132, 2013. (zitiert auf Seite 7, 8 und 10)
- [Myr05] Thomas Myrach. *Temporale Datenbanken in betrieblichen Informationssystemen: Prinzipien, Konzepte, Umsetzung*. Springer-Verlag, 2005. (zitiert auf Seite 12, 13, 14, 16 und 17)
- [RRL⁺05] David SH Rosenthal, Thomas S Robertson, Tom Lipkis, Vicky Reich, and Seth Morabito. Requirements for digital preservation systems: A bottom-up approach. *arXiv preprint cs/0509018*, 2005. (zitiert auf Seite 25)
- [Saa90] Thomas L Saaty. How to make a decision: the analytic hierarchy process. *European journal of operational research*, 48(1):9–26, 1990. (zitiert auf Seite 19)
- [SAA⁺94] Richard Thomas Snodgrass, Ilsoo Ahn, Gad Ariav, Don S Batory, James Clifford, Curtis E Dyreson, Ramez Elmasri, Fabio Grandi, Christian S Jensen, Wolfgang Käfer, et al. Tsql2 language specification. *Sigmod Record*, 23(1):65–86, 1994. (zitiert auf Seite 14)
- [Saa08] Thomas L Saaty. Decision making with the analytic hierarchy process. *International journal of services sciences*, 1(1):83–98, 2008. (zitiert auf Seite xi, 5, 6, 7 und 8)
- [Ste98] Andreas Steiner. *A generalisation approach to temporal data models and their implementations*. PhD thesis, Citeseer, 1998. (zitiert auf Seite 14)
- [uIeV09] Verband Organisations und Informationssysteme e. V. Merksätze des VOI zur revisions sicheren elektronischen Archivierung, 2009. (zitiert auf Seite 25, 26 und 29)
- [vBD06] Marlene van Ballegooie and Wendy Duff. DCC Digital Curation Manual: Instalment on Archival Metadata, 2006. (zitiert auf Seite 28)

- [VER14] SPRINGER GABLER VERLAG. Gabler wirtschaftslexikon. *Stichwort: interne Kommunikation, online im Internet, URL: <http://wirtschaftslexikon.gabler.de/Archiv/326738/interne-kommunikation-v3.html>, Stand, 19:2014, 2014.* (zitiert auf Seite 1)

Hiermit erkläre ich, dass ich die vorliegende Arbeit selbständig verfasst und keine anderen als die angegebenen Quellen und Hilfsmittel verwendet habe.

Magdeburg, den 26.09.16