JOHANNES KEPLER
UNIVERSITÄT LINZ | JKU

SOWI

Sozial- und Wirtschaftswissenschaftliche
Fakultät

Online Analytical Processing mit hetero-homogenen Data Warehouses

Prototypische Implementierung und Benutzerstudie

Diplomarbeit

zur Erlangung des akademischen Grades

Magister der Sozial- und Wirtschaftswissenschaften

(Mag.rer.soc.oec)

im Diplomstudium

Wirtschaftsinformatik

Eingereicht von:
Joachim Roth

Angefertigt am:
Institut für Wirtschaftsinformatik Data- & Knowledge Engineering

Beurteilung: o. Univ.-Prof. DI Dr. Michael Schrefl

Mitwirkung: Mag. Dr. Christoph Schütz

Linz, Juni 2015

Eidesstattliche Erklärung

Ich erkläre an Eides statt, dass ich die vorliegende Diplomarbeit selbstständig und ohne fremde Hilfe verfasst, andere als die angegebenen Quellen und Hilfsmittel nicht benutzt bzw. die wörtlich oder sinngemäß entnommenen Stellen als solche kenntlich gemacht habe.

Die vorliegende Diplomarbeit ist mit dem elektronisch übermittelten Textdokument identisch.

Linz, Juni 2015

_____
Joachim Roth

Abstract

The data which are stored in a data warehouse originate from different, often heterogeneous sources in a company to permanently store these data for data analysis. At present there are two different approaches to consolidate these data, the homogenous and the heterogeneous approach. Using the homogenous approach some information of the heterogeneous data may be lost; using the heterogeneous approach this information can be kept but needs complex data structures and therefore loses clarity. In this thesis we implement a prototype for hetero-homogenous data analysis and we conduct a user-study using the SUS questionnaire (System Usability Scale) in order to evaluate the usability of the hetero-homogenous approach.

Kurzfassung

In einem Data Warehouse werden üblicherweise Daten aus verschiedenen, oftmals heterogenen Quellen eines Unternehmens zusammengeführt um diese langfristig für Datenanalysen zu speichern. Um dies zu bewerkstelligen werden diese Daten mit aktuellen Techniken entweder in einem homogenen oder einem rein heterogenen Schema zusammengeführt. Bei der Überführung in ein homogenes Schema gehen mitunter wichtige Informationen verloren, in einem heterogenen Schema können diese Informationen zwar behalten werden aber die Komplexität der Datenstruktur und Analyse der Daten steigen dadurch mitunter an, darüber hinaus sinkt die Datenqualität. In dieser Arbeit wird der hetero-homogene Ansatz zur Datenanalyse prototypisch implementiert. Eine Benutzerstudie unter Verwendung des SUS-Fragebogens (System Usability Scale) untersucht die Usability des hetero-homogenen Ansatzes.

Inhaltsverzeichnis

1. Einleitung 1
2. Hintergrund 2
2.1. Data Warehouses 2
2.1.1. Multidimensionale Datenmodellierung 2
2.1.2. Grundlagen des OLAP 4
2.2. OLAP-Tool 7
2.3. Benutzerstudien 9
2.3.1. Usability Definition 9
2.3.2. System Usability Scale 9
2.3.3. Single Ease Question 11
3. Hetero-homogenes Data Warehousing 12
3.1. Konzeptuelles Modell 13
3.1.1. M-Objects 14
3.1.2. M-Relationships 15
3.2. Logisches Modell 16
3.2.1. Dimensionen 17
3.2.2. Würfel 18
3.3. Physisches Modell 19
3.3.1. Dimensionstabellen 19
3.3.2. Faktentabellen 21
4. Hetero-homogenes OLAP 24
4.1. Konzept 24
4.2. Implementierung 26
4.2.1. Saiku - Aufbau 26
4.2.2. Saiku - Änderungen 28

5. Studie 36
5.1. Untersuchungsdesign 36
5.1.1. Methoden 36
5.1.2. Testumgebung 38
5.1.3. Durchführung 44
5.2. Ergebnisse 45
5.3. Interpretation 52
6. Zusammenfassung und Ausblick 58
A Appendix 59
Installation 59
Allgemeine Installation und Konfigurationen 59
Installation und Konfiguration heterogenes Data Warehouse 59
Installation und Konfiguration hetero-homogenes Data Warehouse 59
Drehbuch – Anweisungen zu OLAP mit hetero-homogenen DW 61
Drehbuch – Anweisungen zu OLAP mit heterogenen DW 64
Aufgabenstellung und Fragebogen 67
Hetero-homogenes Mondrian-Schema 72
Heterogenes Mondrian-Schema 77
Literaturverzeichnis 80

Abbildungsverzeichnis

Abbildung 1: Alternative Hierarchien in Zeitdimension (vgl. [1] S. 16) 2
Abbildung 2: Dimensionshierarchie mit Dimensionsausprägungen (vgl. [1] S. 18)..... 3
Abbildung 3: Würfel ohne und mit fehlenden Werten (Vgl. [1] S. 22)..... 4
Abbildung 4: Star-Schema mit Fakten- und Dimensionstabellen (Vgl. [1] S. 28)..... 6
Abbildung 5: Saiku mit ausgewähltem Cube „Sales“ 7
Abbildung 6: Saiku nach Auswahl von Ebenen und Kennzahlen..... 8
Abbildung 7: Saiku Filterdialog 8
Abbildung 8: Saiku Filterergebnis 9
Abbildung 9: Würfel mit Verkaufszahlen und Heterogenität in den USA und Mexico 13
Abbildung 10: M-Relationships die einen hetero-homogenen Würfel bilden 15
Abbildung 11: Aktivitätsdiagramm hetero-homogenes Frontend 25
Abbildung 12: Saiku Frontend Struktur..... 27
Abbildung 13: Vergleich der subjektiven Bewertung zu SUS-Score [25]..... 37
Abbildung 14: Testumgebung heterogenes OLAP 39
Abbildung 15: Testumgebung hetero-homogenes OLAP 39
Abbildung 16: Saiku 2.4 nach Änderungen 40
Abbildung 17: Piwik Benutzerlog 40
Abbildung 18: Heterogenes Modell des Würfels für die Studie..... 42
Abbildung 19: Hetero-homogene Hierarchien der Dimension „Store“..... 43
Abbildung 20: Hetero-homogene Hierarchie der Dimension „Customers"..... 43
Abbildung 21: Hetero-homogene Hierarchien der Dimension "Product"..... 43
Abbildung 22: SUS-Score Vergleich hetero-homogen zu heterogen [25]..... 53
Abbildung 23: Hetero-homogene Hierarchien und heterogene Hierarchie Store..... 54
Abbildung 24: Hetero-homogene und heterogene Hierarchie „Product“..... 54
Abbildung 25: Abfrageergebnis in Task 5 ohne Filterung 57

Tabellenverzeichnis

Tabelle 1: Heterogene Dimensionstabelle „Store" 20
Tabelle 2: Homogener USA View für die Dimension "Store" 20
Tabelle 3: Heterogene Faktentabelle „sales_facts_1997“ 21
Tabelle 4: Mittelwert und Median der SUS-Scores und subjektive Bewertung 46
Tabelle 5: Mittelwert und Median der Schwierigkeit der einzelnen Tasks 47
Tabelle 6: Aktionen eines Probanden zu Task 3 in heterogenen OLAP-Tool 55

Diagrammverzeichnis

Diagramm 1: Verteilung der SUS-Scores 46
Diagramm 2: Auswertung subjektive Frage Nutzerfreundlichkeit der Anwendung ... 47
Diagramm 3: Task 1 abgeschlossen 48
Diagramm 4: Task 1 Schwierigkeit 48
Diagramm 5: Task 2 abgeschlossen 49
Diagramm 6: Task 2 Schwierigkeit 49
Diagramm 7: Task 3 abgeschlossen 50
Diagramm 8: Task 3 Schwierigkeit 50
Diagramm 9: Task 4 abgeschlossen 51
Diagramm 10: Task 4 Schwierigkeit 51
Diagramm 11: Task 5 abgeschlossen 52
Diagramm 12: Task 5 Schwierigkeit 52

Auflistung

Auflistung 1: Hierarchie aus M-Objects im HH-DW Project 14
Auflistung 2: M-Relationships im Hetero-Homogeneous Data Warehouse Project .. 16
Auflistung 3: Dimension Store in Mondrian-Schema 17
Auflistung 4: Mondrian-Schema mit <VirtualCube> 19
Auflistung 5: Queries zur Erzeugung der Hierarchien für die "Store" Dimension 20
Auflistung 6: Mondrian Hierarchie für Geschäfte in den USA 20
Auflistung 7: Sicht minimal homogener Sales Würfel 21
Auflistung 8: Sicht auf Würfel mit Geschäften und Kunden aus den USA 22
Auflistung 9: Mondrian-Schema Ausschnitt mit Customers Dimension 22
Auflistung 10: Hierarchien-Namen Dimension „Product" 28
Auflistung 11: Kennzahlen "Store Sales" 28
Auflistung 12: Verändertes Template „template-measures" in index.html 29
Auflistung 13: Saiku „QUERY_PROPERTIES" in Settings.js 29
Auflistung 14: Teile der Funktion "select_measure" in "DimensionList.js" 30
Auflistung 15: Funktionen in Query.js 30
Auflistung 16: Funktion „put" in QueryAction.js 31
Auflistung 17: Initialisierung von HeteroHomogenousRuleset 32
Auflistung 18: Aufnehmen von Ebenen und Kennzahlen in "update_hh" 32
Auflistung 19: Erstellung der Abfrage für Webservice 34
Auflistung 20: Austausch von Ebenen 34
Auflistung 21: Filter neu setzen 35
Auflistung 22: Piwik Trackingcode 41
Auflistung 23: Tracking einer Kennzahl bei Aktion "ondrag" oder "onclick" 41

1. Einleitung

In einem Data Warehouse werden üblicherweise Daten aus verschiedenen, meist heterogenen Datenquellen zusammengeführt und langfristig gespeichert um so eine Datenanalyse durchführen zu können und daraus betriebswirtschaftliche Entscheidungen zu treffen. Die Zusammenführung dieser Daten führt, mit den derzeit gängigen Techniken, in ein homogenes multidimensionales Schema, dadurch können je nach Unterschied in den heterogenen Datenquellen Informationen verloren gehen (Kennzahlen, Granularität der Kennzahlen, Hierarchien, Ebenen, usw.). Werden die heterogenen Daten nicht in einem einzigen homogenen Schema zusammengeführt sondern in ein vollständig heterogenes, können diese Informationen behalten werden, allerdings wird dadurch die Analyse je nach Menge an unterschiedlichen Daten schwieriger und die Datenqualität leidet. Dies ist gerade in einem Marktumfeld mit ständig steigender Quantität an Daten und hoher Mitarbeiterfluktuation wie sie in der heutigen Zeit besteht, problematisch da auf Grund dessen häufig Wissen zu den im Unternehmen bestehenden Daten verloren geht. Mit dem hetero-homogenen Ansatz [2] können heterogene Daten teilweise abgebildet werden indem hetero-homogene Sub-Hierarchien eingeführt werden die ein gemeinsames minimales homogenes Schema verwenden aber in der jeweiligen Sub-Hierarchie heterogen sind. Durch diese Heterogenität können Sub-Cubes mit zusätzlichen Kennzahlen, verfeinerten Fakten und unterschiedlichen Kennzahleneinheiten (z.B. Währung) gebildet werden. Dadurch soll die Datenqualität erhöht werden ohne Informationen zu verlieren. Dieses Konzept wurde bereits als Prototyp in Oracle umgesetzt [3]. Um den hetero-homogenen Ansatz auch in gängigen Anwendungen für Business Intelligence (BI) ohne zu großen Aufwand (wieder)verwenden zu können wird von Schütz et al. [4] ein Leitfaden vorgelegt um im Common Warehouse Metamodel (CWM - Standard für Data Warehouse Metadaten) hetero-homogene Data Warehouses zu modellieren. Die Vorteile für den Anwender eines Tools für Online Analytical Processing (OLAP) die der hetero-homogene Ansatz bringen soll und ob diese Anwender mit dem komplexeren Suchvorgang umgehen können, wurde allerdings bisher noch nicht überprüft. In dieser Diplomarbeit soll zuerst ein bestehendes OLAP-Tool um bestimmte hetero-homogene Fähigkeiten erweitert werden und anschließend, mit Hilfe einer Benutzerstudie, untersucht werden ob Anwender ohne Erfahrung mit dem hetero-homogenen Ansatz umgehen können. Von Interesse dabei ist in erster Linie die Usability und dabei im Speziellen die Komplexität des Suchvorgangs. Die Studie ist so aufgebaut, dass eine Gruppe bestimmte Aufgaben mit einem hetero-homogenen Data Warehouse löst und eine zweite Gruppe zum Vergleich die gleichen Aufgaben mit einem rein heterogenen. Die beiden Gruppen verwenden das gleiche OLAP-Tool. Nach Beendigung aller Aufgaben müssen die Benutzer einen SUS-Fragebogen (System Usability Scale) beantworten, um feststellen zu können, wie die beiden unterschiedlichen Anwendungen, bei gleichen Aufgaben, wahrgenommen werden. Außerdem müssen die Benutzer nach Beendigung jeder Aufgabe, inklusive der darin enthaltenen Unteraufgaben, die Schwierigkeit dieser beurteilen. Die Beurteilung erfolgt dabei mittels einer SEQ (Single Ease Question) genannten siebenstufigen Likert-Skala [5]. Nicht geprüft wird ob durch den hetero-homogenen Ansatz bessere ökonomische Entscheidungen getroffen werden.

In Kapitel 2 werden die für diese Arbeit relevanten Hintergrundinformationen behandelt, dabei wird in Kapitel 2.1 Data Warehousing allgemein besprochen, in Kapitel 2.2 wird das Open-Source OLAP-Frontend Saiku (Version 2.4) vorgestellt und Kapitel 2.3 beschäftigt sich mit Usability und den Methoden, die in dieser Arbeit verwendet werden um diese zu prüfen. Kapitel 3 beschreibt zunächst kurz die Grundprinzipien des hetero-homogenen Ansatzes und zeigt anschließend die Modellierung von hetero-homogenen Data Warehouses mit Mondrian 3. Kapitel 4 beschreibt das Konzept und die Adaptierung des OLAP-Frontends Saiku 2.4 um hetero-homogene Aspekte, die in Kapitel 3 besprochen wurden. Der Aufbau, die Durchführung und die Ergebnisse der Benutzerstudie werden in Kapitel 5 dargestellt, in Kapitel 6 folgen die Zusammenfassung und ein Ausblick für zukünftige Arbeiten in diesem Bereich.

2. Hintergrund

In diesem Kapitel werden die Grundlagen zum Thema Data Warehousing, OLAP und Benutzerstudien besprochen. Außerdem wird das OLAP-Frontend Saiku in der Version 2.4 welches als Grundlage für die Implementierung des Prototyps in dieser Arbeit verwendet wurde, vorgestellt.

2.1. Data Warehouses

William H. Inmon definiert ein Data Warehouse als eine themenorientierte, integrierte, permanente und zeitbezogene Datensammlung aus unterschiedlichen Quellen eines Unternehmens, die die Entscheidungsfindung im Management unterstützen soll.

"A data warehouse is a subject-oriented, integrated, nonvolatile, and time-variant collection of data in support of management's decisions. The data warehouse contains granular corporate data." ([6] S. 31).

Farkisch ([1] S. 5f) erklärt die Definition wie in folgendem Absatz beschrieben.

Mit themenorientiert ist gemeint, dass in einem Data Warehouse nicht einzelne Transkationen, Buchungen und Aufträge relevant sind sondern Kennzahlen wie Umsatz, Kosten, Menge usw. Hier ist es also die Aufgabe der Datenbank, einen spezifischen Themenbereich abzubilden. Dazu müssen die Daten sowohl physisch als auch logisch organisiert werden, um eine multidimensionale Sichtweise mit einer Unterscheidung von quantifizierbaren Kennzahlen und beschreibenden Informationen zu ermöglichen. Dass die Daten in einem Data Warehouse integriert sind bedeutet, dass sie aus verschiedenen Quellen und Fremdsystemen stammen aber über bestimmte Prozesse vereinheitlicht werden um für Abfragen verwendbar zu sein. Mit permanent, bzw. nicht flüchtig, ist gemeint, dass Daten die einmal in das Data Warehouse gebracht wurden nicht modifiziert oder verändert werden um zeitbezogene Abfragen darüber ausführen zu können (z.B. Wie hoch war der Umsatz des Unternehmens im Monat August).

In diesem Unterkapitel werden die Grundlagen der multidimensionalen Datenmodellierung und Grundlagen des OLAP beschrieben, falls nicht anders angegeben beziehen sich diese Angaben auf die Ausführungen von Farkisch [1].

2.1.1. *Multidimensionale Datenmodellierung*

Bei der Datenmodellierung in Data Warehouses hat sich die multidimensionale Datenmodellierung durchgesetzt und im Einsatz bewährt, da diese betriebswirtschaftliche Fragestellungen, die von Natur her multidimensional sind, sehr gut abbildet und Abfragen darauf gut bedienen kann. In diesem Unterkapitel werden die wichtigsten Termini für die multidimensionalen Datenmodellierung kurz besprochen.

Dimension

Bezogen auf multidimensionale Datenmodellierung sind Dimensionen Datenstrukturen, durch die verschiedene Aspekte der zu analysierenden Daten dargestellt werden sollen (z.B. Zeit, Ort, Produkt, Verkäufer, Kunde). Durch Dimensionselemente in einer Dimension wird diese feingranular unterteilt. Dabei gliedert sich die Dimension in hierarchisch angeordnete Ebenen (siehe Abbildung 1). Aus Anwendersicht sind diese Hierarchien Bestandteile der Dimension. In einer Dimension können sich auch mehrere alternative Hierarchien befinden, die unter-

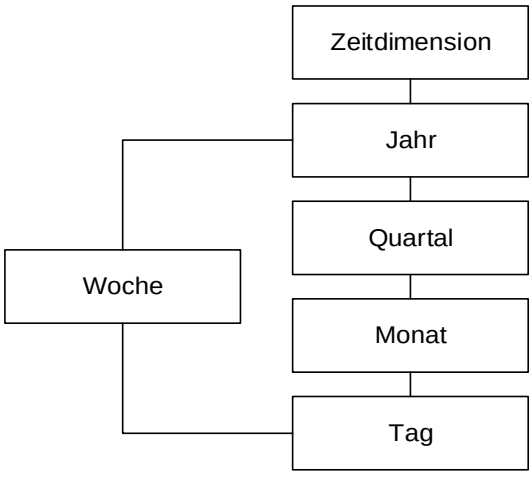


Abbildung 1: Alternative Hierarchien in Zeitdimension (vgl. [1] S. 16)

schiedliche und unabhängige Möglichkeiten der Gruppierung von Daten in einer Dimension darstellen. Die Ebenen der Dimension werden durch die Klassifizierung der Dimensionselemente gebildet, diese Elemente einer Ebene fassen Teilmengen der nächstniedrigeren Stufe zusammen. Bildhaft gesprochen sind diese Dimensionselemente also Blätter eines Baumes, der als Klassifikationshierarchie bezeichnet wird. Da Klassifikationshierarchien, durch eine große Anzahl von Knoten und Verknüpfungen, äußerst umfangreich werden können, werden Dimensionen über das Schema ihrer Klassifikationshierarchie dargestellt und die Klassifikationsknoten werden durch ihre Klassifikationsebenen repräsentiert. Die Beziehungen zwischen den Klassifikationsebenen in einer Hierarchie werden in relationalen Datenbanken durch funktionale Abhängigkeiten abgebildet. Die Klassifikationsebene beschreibt also den Verdichtungsgrad innerhalb ihrer Hierarchie (Vgl. [1] S. 13ff). Abbildung 2 zeigt eine Dimensionshierarchie mit ihren Ebenen und deren Ausprägungen.

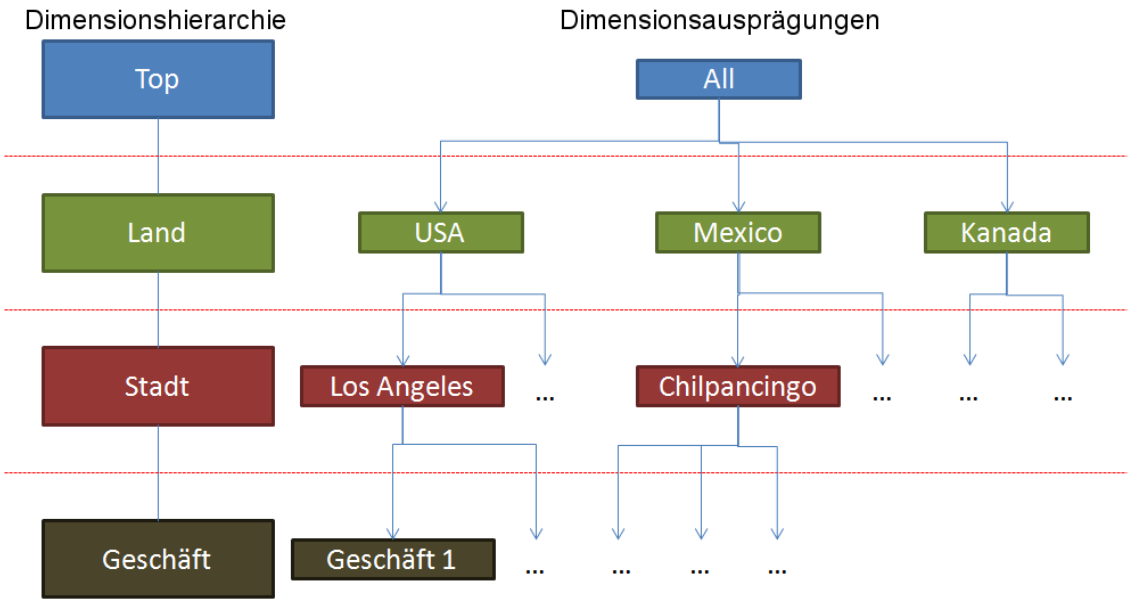


Abbildung 2: Dimensionshierarchie mit Dimensionsausprägungen (vgl. [1] S. 18)

Fakten und Kennzahlen

Dimensionen und ihre Hierarchien stellen den qualifizierenden Teil eines multidimensionalen Datenmodells dar, der quantitative Teil wird durch Fakten und Kennzahlen dargestellt. Dies sind die Zellen bzw. die Inhalte dieser Zellen in einem Würfel dessen Kanten durch Dimensionen dargestellt werden. Kennzahlen sind betriebswirtschaftliche Sachverhalte wie Umsatz, Gewinn oder Kosten usw. und entstehen durch die Anwendung von arithmetischen Operationen oder werden aus anderen Kennzahlen und Fakten abgeleitet. Formal wird also eine Kennzahl M (Measure) durch diese Komponenten definiert (Vgl. [1] S. 19f):

- Granularität *G*
- Berechnungsformel *f()* über eine **nicht leere** Teilmenge aller im multidimensionalen Schema enthaltenen Fakten *F*
- Summationstyp

$$M = (G, f(F_1, F_2,\ldots, F_k), SumTyp)$$

$$= ((G_1, G_2,\ldots, G_k), f(F_1, F_2,\ldots, F_k), SumTyp).$$

Nicht alle Fakten und Kennzahlen können aggregiert werden, dazu wird nämlich die Summierbarkeit, also die inhaltliche Korrektheit der Anwendung einer Aggregationsfunktion auf einen Würfel vorausgesetzt. Summierbarkeit verlangt Disjunktheit, Vollständigkeit und Typverträglichkeit. Mit Disjunktheit ist gemeint, dass der Wert einer Kennzahl nur genau einmal in die Berechnung eines Ergebnisses ein-

geht. Vollständigkeit sagt aus, dass Kennzahlen auf höherer Aggregationsstufe aus Werten aus tieferer Ebene berechnet werden. Typverträglichkeit schließlich besagt, dass Kenngrößen, die ein Ereignis wie Verkäufe beschreiben, in allen Dimensionen aggregiert werden können. Im Gegensatz dazu dürfen Kenngrößen, die einen Zustand zu einem bestimmten Zeitpunkt repräsentieren, nicht bezüglich der Zeitdimension summiert werden ([1] S. 20). Eine umfassende Übersicht über Summierbarkeitsprobleme gibt Mazón et al. [7].

Würfel

Ein Würfel ist eine multidimensionale Darstellung von einer oder mehreren Kennzahlen, er wird durch eine Kombination aus Dimensionen, die die Achsen des Würfels bilden, aufgespannt. Der so konstruierte Würfel enthält also Datenzellen die durch die einzelnen Dimensionen (z.B. Ort, Zeit, Produkt) und Kennzahlen gebildet werden. Ein Würfel kann beispielsweise durch das Ändern der ausgewählten Hierarchieebene einer Dimension aggregiert oder verfeinert (Roll-up & Drill Down) werden. Der Würfel kann gedreht werden, sodass eine andere Kombination aus Dimensionen sichtbar wird. Es können einzelne Scheiben des Würfels oder generelle bestimmte spezielle Teile des Würfels, durch Filterung auf bestimmte Elemente, in einer oder mehreren Dimensionen gebildet werden (Slice & Dice), dabei bildet das neue Ergebnis wieder einen Würfel (Vgl. [8] S. 102ff). In der Realität kommt es häufig vor, dass nicht alle Zellen eines Würfels Werte aufweisen. Dies kann entweder auf einen Fehler im Ladevorgang zurückgeführt werden, wodurch der Würfel nicht vollständig aufgebaut werden konnte, was zu falschen Ergebnissen führt, oder aber darauf, dass ein bestimmtes Ereignis nicht möglich ist, nicht bekannt ist oder nicht eintreten kann. Beispielsweise eine Abfrage über ein Produkt in einer bestimmten Filiale, in der dieses gar nicht im Sortiment ist und somit tatsächlich gar nicht angeboten werden kann und damit kein Ergebnis hat. Es würde aber die Information geben, dass dieses Produkt in der vorgegeben Filiale nicht verkauft wurde was einem numerischen Wert von 0 entspricht. So wäre also, logisch betrachtet der Würfel voll besetzt und die weiter oben beschriebenen Operationen finden mit allen Zellen statt. Daher müssen klare Regeln existieren wie leere Zellen bzw. fehlende Werte in einem Würfel behandelt werden. Auf der Ebene der Modellierung können nicht existierende Werte entweder durch NULL-Werte oder durch eine numerische Null (0) realisiert werden. [1] S. 21f

Abbildung 3 zeigt exemplarisch einen vollständig gefüllten Würfel und einen zweiten Würfel mit leeren Zellen.

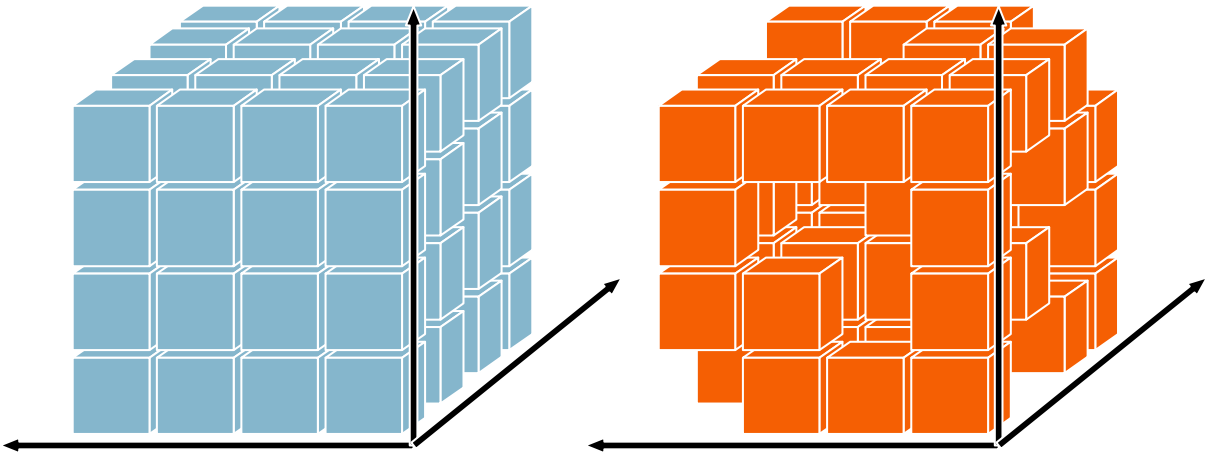


Abbildung 3: Würfel ohne und mit fehlenden Werten (Vgl. [1] S. 22)

2.1.2. Grundlagen des OLAP

OLAP steht für Online Analytical Processing (OLAP) und wird im Gabler Wirtschaftslexikon so definiert: *„Konzept für die im Dialogbetrieb realisierte Verdichtung und Darstellung von managementrelevanten Daten aus einem Data Warehouse. Bei den Daten handelt es sich um Faktendaten, wie z.B. Umsatz- oder Kostenkennzahlen, denen eine Vielzahl von Dimensionsmerkmalen zugeordnet ist, wie z.B. Merkmale von Regionen, Produkten oder Zeiträumen, für die die Kennzahlen angefallen sind.*

Diese Dimensionen sind in Hierarchien angeordnet, d.h. es gibt über- und untergeordnete Regionen, Produktgruppen und Zeiträume. OLAP-Software unterstützt v.a. die Aggregation von Kennzahlen über bestimmte Dimensionen und Hierarchieebenen sowie die grafische Darstellung der Kennzahlen" [9].

Entsprechend der Definition bieten OLAP-Tools generell Funktionen für einen Benutzer um flexible, benutzerfreundliche Ad-hoc-Analysen über einen bestimmten Datenraum ausführen zu können. Wie frei der Benutzer dabei vorgehen kann und wie die Darstellung der Daten, Dimensionen, Kennzahlen usw. aussieht ist dabei von den einzelnen OLAP-Tools selbst abhängig ([8] S. 106)

E. F. Codd formulierte 1993 zwölf Regeln für OLAP-Tools und erweiterte diese 1995 um sechs weitere.

Codds zwölf original Regeln (Vgl. [10]):

1. **Multidimensionale, konzeptionelle Sichten** (Sicht auf verschiedene Kennzahlen, von verschiedenen Dimensionen aus, muss möglich sein).
2. **Transparenz** (Datenquellen, Werkzeuge und deren zugrundeliegende Technologien sollen den Benutzer nicht in seiner Produktivität beeinträchtigen).
3. **Flexible Zugriffsmöglichkeiten** (interne- und externe Quellen).
4. **Konsistente Effizienz** in der Berichtserstellung (unabhängig der Datenmenge und Anzahl der Dimensionen müssen Berichte ohne Leistungsabfall erstellt werden können).
5. **Client/Server Architektur.**
6. **Generische Dimensionen** (wird eine Dimension in ihrer Funktionalität erweitert muss diese Erweiterung auch für alle anderen Dimensionen gelten).
7. **Automatische Anpassung auf physischer Ebene** (OLAP-Tools sollen ihre physischen Schemen automatisch anpassen um unterschiedliche Datenmodelle usw. zu integrieren).
8. **Mehrbenutzerunterstützung** (gleichzeitiger Datenbankzugriff, Zugriffsautorisierung, Sicherheits- und Integritätsmechanismen).
9. **Uneingeschränkte kreuzdimensionale Operationen** (Ableitungen und Berechnungen zwischen und über Dimensionen müssen automatisch von dem OLAP-Tool durchgeführt werden).
10. **Intuitive Datenbearbeitung** (Usability).
11. **Flexible Berichterstellung** (beliebiges Pivotieren der Daten für den Bericht).
12. **Unlimitierte Dimensions- und Aggregationsebenen.**

Codds sechs zusätzliche Regeln:

13. **Datenintegration** (Transparente Überführung und Integration von Daten aus Vorsystemen).
14. **Unterstützung von verschiedenen Analysemodellen** (Slicing, Dicing, Drill-down Operationen, „Was wäre, wenn…"-Fragen usw.).
15. Trennen der **Analyse-orientierten Daten** von den **Daten des Quellsystems** (Veränderung von Daten darf die operativen Daten nicht beeinflussen).
16. Speichern der **OLAP-Resultate getrennt** von den **Eingabedaten** (OLAP-Tools können Resultate modifizieren, das darf sich nicht auf das Data Warehouse durchschlagen).
17. Unterscheidung von **leeren Feldern (NULL)** und Feldern **mit numerischer Null (0)** muss möglich sein.
18. Behandlung von **leeren Feldern** (das OLAP-Tool muss leere Felder effizient verwalten können).

Eine einfachere, aus den fünf Schlagwörtern *„Fast Analysis of Shared Multidimensional Information (FASMI)"* bestehende, Definition für OLAP-Tools kommt von Pendse und Creeth [11]:

- **Fast** bedeutet, dass die meisten Anfragen innerhalb von fünf Sekunden und komplexe Anfragen in maximal 20 Sekunden beantwortet sein müssen.

- **Analysis** meint, dass der Anwender in die Lage versetzt wird ohne Programmierung, Ad-hoc Analysen durchführen zu können.
- **Shared** verlangt die Mehrbenutzerkontrolle wie sie ein übliches RDBMS bietet.
- **Multidimensional** meint, dass unabhängig der zugrundeliegenden Datenstruktur eine konzeptionelle multidimensionale Sicht umgesetzt werden muss.
- **Information** ist das Resultat.

Die Speicherung und Verwaltung von multidimensionalen Daten verlangt, dass dies entweder in relationalen Strukturen umgesetzt wird, direkt in multidimensionalen Strukturen und Arrays oder in hybriden Typen, die beides vereinen. In dieser Arbeit wird nur der rein relationale Ansatz besprochen, da die Umsetzung des Prototyps diesen verwendet.

OLAP-Tools, die mit Daten in relationalen Strukturen umgesetzt sind, werden relationales OLAP oder auch ROLAP genannt. Die Datenhaltung in einem RDBMS erlaubt eine effiziente Umsetzung und Verarbeitung von multidimensionalen Anfragen. Die wichtigste relationale Datenstruktur in diesem Zusammenhang wird Star-Schema genannt und in bis zu 2^d Tabellen in denormalisierter Form (zweite Normalform ist erfüllt aber nicht die dritte) abgebildet, ‚d' steht dabei für die Anzahl der Dimensionen. Das Star-Schema hat zwei Tabellentypen, nämlich die Dimensionstabelle und die Faktentabelle, es trägt den Namen „Star" auf Grund der grafischen Anordnung, bei der die Dimensionstabellen in Sternform um die Faktentabelle angeordnet sind, wie in Abbildung 4 dargestellt (Vgl. [1] S. 27).

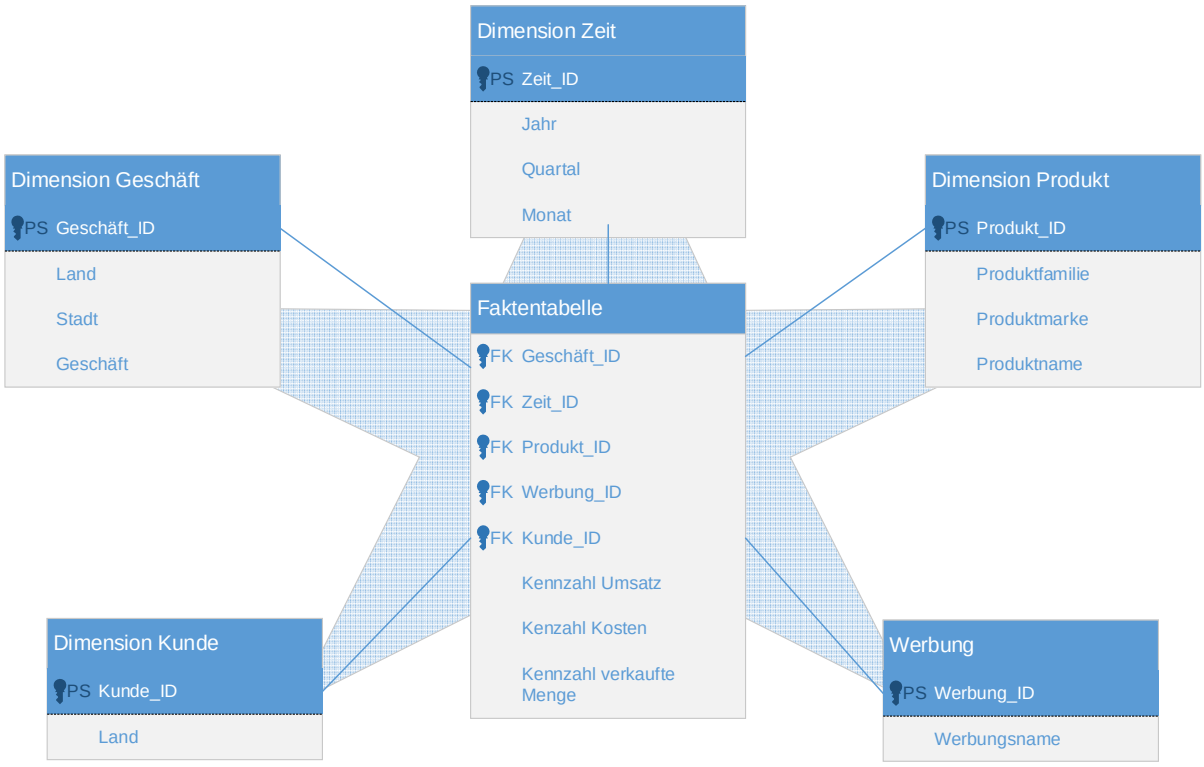


Abbildung 4: Star-Schema mit Fakten- und Dimensionstabellen (Vgl. [1] S. 28)

Wie vorhin angesprochen, sind die Dimensionshierarchien im Star-Schema denormalisiert, die Relationen sind also, auf Grund von transitiven Abhängigkeiten zwischen Attributen von Dimensionstabellen, nur bis zur zweiten Normalform normalisiert. Liegen jedoch Relationen in dritter Normalform vor, müssen abhängige Attribute einer Dimension in weitere kleinere Tabellen ausgelagert werden, um transitive Abhängigkeiten aufzulösen. In diesem Fall ist eine spezielle Variante des Star-Schemas vorhanden, die Snowflake-Schema genannt wird. Des Weiteren existieren Mischformen, in denen einzelne Dimensionstabellen in dritter Normalform und andere in zweiter Normalform vorhanden sind. Außerdem besteht die Möglichkeit, dass in einem Schema mehrere Faktentabellen vorhanden sind

und somit eine Galaxie bilden (vgl. [1] S. 30ff), diese Fälle werden hier nicht mehr genauer beschrieben.

2.2. OLAP-Tool

Da diese Arbeit vorsieht eine bestehende ROLAP-Anwendung zu adaptieren und darauf eine Benutzerstudie durchzuführen, kommen nur weit fortgeschrittene Open-Source-Projekte in Frage. Dabei hat sich, unter Berücksichtigung dieser Vorgaben und der obigen Definition und Anforderungen an OLAP-Tools, die in diesem Kapitel beschriebene Anwendung Saiku als adäquates Frontend herauskristallisiert.

Saiku ist ein sehr weit fortgeschrittenes, mittlerweile auch kommerziell genutztes, leichtgewichtiges OLAP-Frontend, das sich im Vergleich zu anderen Open-Source-Projekten besonders durch seine intuitive Benutzbarkeit auszeichnet und unter der Apache Software License veröffentlicht wurde. Aus diesem Grund wurde Saiku für diese Arbeit ausgewählt. In diesem Kapitel werden zunächst kurz die wichtigsten Arbeitsschritte beschrieben, die mit Saiku (Version 2.4) durchgeführt werden können und anschließend kurz wichtige technische Informationen zum System dargestellt.

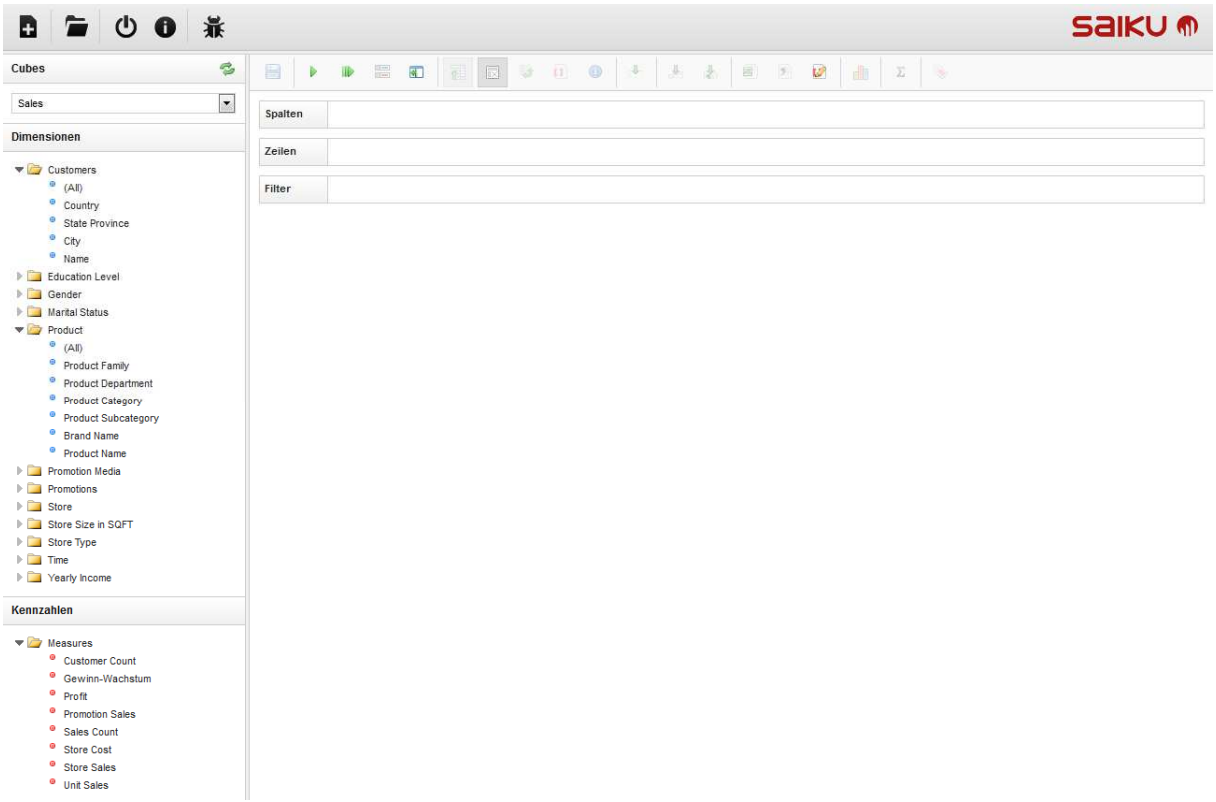


Abbildung 5: Saiku mit ausgewähltem Cube „Sales“

In Abbildung 5 ist Saiku in der Version 2.4 zu sehen. Dabei ist ein Cube mit dem Namen „Sales“ ausgewählt, für den mehrere Dimensionen und Kennzahlen verfügbar sind. Die Dimensionen werden als Ordner dargestellt. Sind diese Ordner aufgeklappt, werden die jeweiligen Hierarchien auf die in dieser Dimension aufgerollt werden kann gezeigt (in Abbildung 5 sind die Hierarchien der Dimensionen „Customers“ und „Product“ zu sehen). Die Kennzahlen befinden sich unterhalb der Dimensionen. Die Dimensionen bestimmen den örtlichen, zeitlichen und sachlichen Gültigkeitsbereich der Kennzahlen. Dimensionen bestehen aus mehreren Ebenen, die hierarchisch geordnet sind. Die hierarchische Ordnung der Ebenen erlaubt die Aggregation von Kennzahlen nach den verschiedenen Ebenen. Durch Klicken oder per Drag & Drop können einzelne Ebenen und Kennzahlen für eine Abfrage ausgewählt werden. Abbildung 6 zeigt das Ergebnis nachdem die Ebene „Country“ aus der Dimension „Customers“, die Ebene „Product Family“ aus der Dimension „Product“ und die Kennzahl „Profit“ auf die Spalten- und Zeilenfelder gezogen wurden und die Abfrage für diese Auswahl ausgeführt wurde.

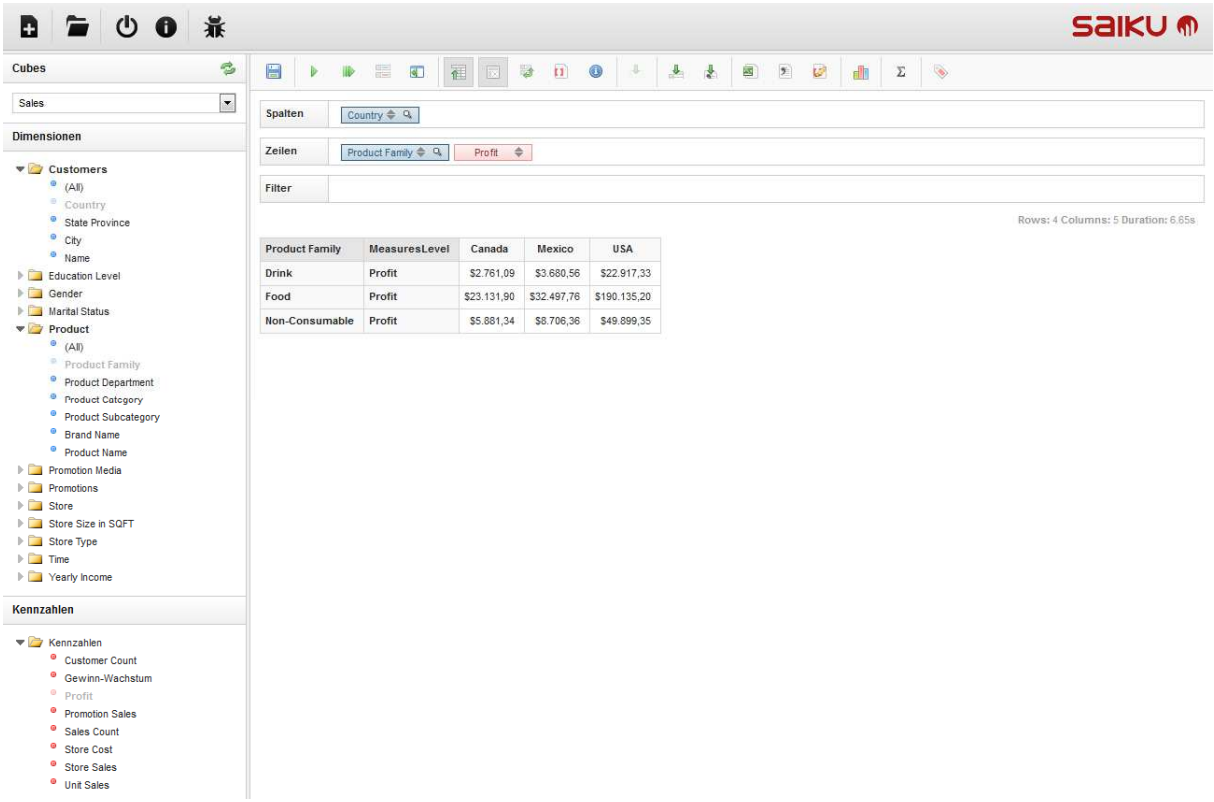


Abbildung 6: Saiku nach Auswahl von Ebenen und Kennzahlen

Sind beispielsweise nur Kunden aus den USA relevant für die Abfrage, kann durch einen Klick auf das Lupensymbol einer ausgewählten Ebene - in diesem Beispiel der auf Spalten abgelegten Ebene „Country“ - ein Filterdialog wie in Abbildung 7 dargestellt aufgerufen und ein Filter gesetzt werden.

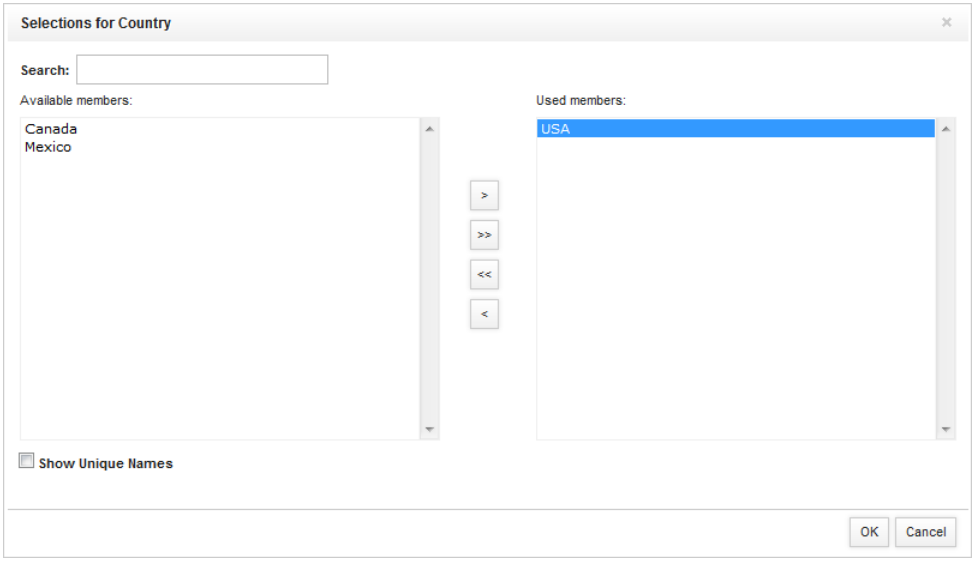
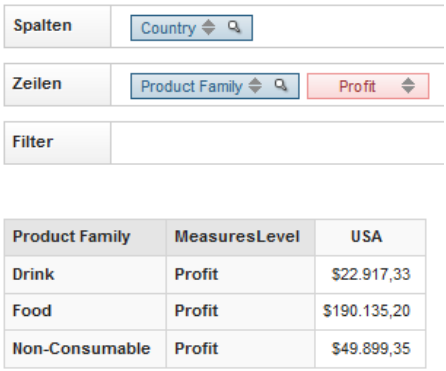


Abbildung 7: Saiku Filterdialog

Abbildung 8 zeigt das Ergebnis nachdem der Filter gesetzt und die Abfrage erneut gestartet wurde. Saiku erlaubt auch sonstige grundlegende OLAP Operationen auszuführen, die Ergebnisse grafisch in Form verschiedener Diagramme darzustellen, Abfragen zu speichern, selbst Abfragen mit der für OLAP Datenbanken etablierten Abfragesprache MDX zu formulieren und besitzt einige weitere Features wie das Exportieren der Abfrageergebnisse in eine CSV Datei usw. Für einen detaillierteren Überblick der Möglichkeiten aus Anwendersicht wird, da für diese Arbeit nicht relevant, auf die Saiku Dokumentation [12] verwiesen.



Product Family	MeasuresLevel	USA
Drink	Profit	$22.917,33
Food	Profit	$190.135,20
Non-Consumable	Profit	$49.899,35

Abbildung 8: Saiku Filterergebnis

Saiku ist sowohl als Plugin für die Business Intelligence Plattform Pentaho als auch als eigene Stand-Alone Version verfügbar und verwendet dafür standardmäßig die olap4j API zur Kommunikation mit Mondrian 3. Dementsprechend ist das Backend in Java implementiert und läuft auf einem Apache Tomcat Webserver. Das Frontend ist eine Single-page-Webanwendung und wurde unter der Verwendung des Javascript Frameworks Backbone das einer MV* Designpattern folgt [12], implementiert.

2.3. Benutzerstudien

Wie bereits in der Einleitung erwähnt, geht es bei der Benutzerstudie in dieser Arbeit darum zu evaluieren, wie der hetero-homogene Ansatz in OLAP Anwendungen in Bezug auf Usability bei Anwendern abschneidet. Aus diesem Grund wird hier zunächst erörtert wie sich Usability definiert und anschließend werden die Methoden, ein quantitativer Fragebogen, der sich genau für solche Aufgaben als bewährt [13] erwiesen hat und bereits seit 1986 eingesetzt wird und eine einfache Frage zur Bewertung der Schwierigkeit einzelner Aufgaben [14] vorgestellt, die in dieser Arbeit verwendet werden.

2.3.1. Usability Definition

Usability beschreibt die allgemeine Qualität eines Systems und bestimmt sich durch die Gebrauchstauglichkeit die aus Benutzungseffizienz und Zufriedenstellung des Anwenders besteht und ist somit stark kontextabhängig [15]. Die DIN EN ISO 9241-11 definiert die Gebrauchstauglichkeit als *„das Ausmaß, in dem ein Produkt von bestimmten Benutzern in einem bestimmten Nutzungskontext genutzt werden kann, um bestimmte Ziele effektiv, effizient und zufriedenstellend zu erreichen“* [16]. Dieser Nutzungskontext hängt von diesen Punkten ab [16]:

- Benutzern
- Arbeitsaufgaben
- Arbeitsmitteln (Hardware, Software, Materialien)
- der physischen und sozialen Umgebung

In [15] werden effektiv, effizient und zufriedenstellend, im Sinne der Usability so beschrieben:

- Die **Effektivität** ist die Genauigkeit und Vollständigkeit mit der ein Benutzer ein bestimmtes Ziel erreichen kann.
- Die **Effizienz** wird charakterisiert durch den Aufwand der zur Erreichung des Ziels angewandt werden muss.
- Die **Zufriedenheit** ist ein subjektives Benutzerkriterium, bezogen auf die Verwendung des verwendeten Systems

2.3.2. System Usability Scale

Die System Usability Scale oder auch SUS genannt, wurde für diese Arbeit zur Evaluierung der Usability des hetero-homogenen Ansatzes ausgewählt, da sie ein frei verfügbarer und bereits sehr lange

etablierter Fragebogen zu einer Quantitativen Erhebung von Usability über den gesamten Nutzungskontext hinweg und dabei sehr einfach und schnell durchführbar ist. Außerdem ist dieser Fragebogen sehr gut dazu geeignet, Produkte in verschiedenen Varianten und Versionen zu vergleichen, bzw. im Entwicklungsprozess iterativ zu kontrollieren wie weit sich das Produkt verbessern konnte und sich die Usability geändert hat. Des Weiteren ist der Fragebogen technologieagnostisch. Er kann also beispielsweise für Hardwaresysteme und Softwaresysteme gleichermaßen eingesetzt werden und ist, laut Studien von [17], überaus zuverlässig und sowohl für Studienteilnehmer leicht zu beantworten, als auch von Studienleitern einfach auszuwerten. Anzumerken ist, dass bei ähnlich hohen Werten für zwei Systeme nur eine begrenzte Aussagekraft vorhanden ist aber Tendenzen erkennbar sind. Der Fragebogen wurde von seinem Entwickler selbst als *„a quick and dirty usability scale“* [15] bezeichnet, weist allerdings bereits bei einer kleinen Anzahl an Probanden (ab 12) eine sehr hohe Reliabilität auf. Selbst bei noch weniger Probanden (8-10) ist der SUS-Fragebogen akkurater als andere bekannte Fragebögen die zum Testen von Systemen auf ihre Usability genutzt werden (QUIS, CSUQ, Words) [13].

Aufgebaut ist der Fragebogen dabei aus zehn positiv-negativ alternierenden Fragen, mit einer fünfgliedrigen Antwortskala im Likert-Format, die von „strongly-agree“ bis „strongly-disagree“ reicht. Ausgewählt wurden diese zehn Fragen aus einem größeren, aus fünfzig Fragen bestehenden Block der zuvor zwanzig Probanden vorgelegt wurde, um zwei Softwaresysteme zu vergleichen, wobei eines der Systeme einfach zu bedienen war und das zweite extrem schwierig. Die zehn für den SUS-Fragebogen ausgewählten Fragen waren jene, die die stärksten Reaktionen ausgelöst haben, wobei eine Hälfte der Fragen mit starker positiver Reaktion („strongly-agree“) und die andere Hälfte mit starker negativer Reaktion („strongly-disagree“) gewählt wurden um sicherzustellen dass, Probanden beim Durchlesen der Fragen wirklich darüber nachdenken müssen, ob sie dieser zustimmen oder sie ablehnen [15].

Im Original von Brooke lauten die zehn Fragen:

1. I think that I would like to use this system frequently
2. I found the system unnecessarily complex
3. I thought the system was easy to use
4. I think that I would need the support of a technical person to be able to use this system
5. I found the various functions in this system were well integrated
6. I thought there was too much inconsistency in this system
7. I would imagine that most people would learn to use this system very quickly
8. I found the system very cumbersome to use
9. I felt very confident using the system
10. I needed to learn a lot of things before I could get going with this system

Die SUS-Score, die einen Wert für die Usability gibt ist eine Zahl die sich aus der Summe der Werte aller Fragen ergibt. Es ist aber zu berücksichtigen, dass der Wert einer Frage alleine keine Aussagekraft hat, nur die Summe aller zehn Fragen zusammen. Um die SUS-Score zu berechnen werden den Antworten jeder Frage Werte von 0 – 4 zugewiesen. Für die positiv formulierten Fragen hat dabei die Antwort „strongly-agree“ den Wert 4 usw. bis zu „strongly-disagree“ mit 0. Für die negativ formulierten läuft dies genau in die andere Richtung, „strongly agree“ hat also den Wert 0 und „strongly-disagree“ den Wert 4. Die Summe dieser Werte muss anschließend mit 2,5 multipliziert werden, um die Score die folglich in einem Bereich von 0 bis 100 sein kann, zu erhalten. 100 stellt dabei das bestmögliche Ergebnis dar [15].

2.3.3. Single Ease Question

Da der zuvor beschrieben SUS-Fragebogen zwar ein sehr gutes Instrument ist um generell die Usability eines Systems zu testen aber nur wenig über einzelne Aufgaben, die mit dem geprüften System gelöst werden können, aussagt, wurde für diese Arbeit noch eine weitere Möglichkeit gesucht um die Zufriedenheit von Anwendern mit dem System, für einzelne Aufgaben zu messen. Sauro und Dumas [14] untersuchten genau für solche Zwecke drei einfache „One-Question, Post-Task Usability Questionnaires“ und beschreiben eine einfache, für Probanden schnell auszufüllenden siebenstufige Likert-Skala Frage als sehr sichere und mit anderen Kennzahlen korrelierende Messgröße. Beispielweise korreliert diese Bewertung mit der Zeit die bis zur Fertigstellung des Tasks benötigt wird oder ob dieser erfolgreich abgeschlossen wurde. Für die Bewertung der Aufgabe wird dabei diese Frage gestellt:

„Overall, this task was:”

	1	2	3	4	5	6	7	
Very difficult								Very easy

Da diese Bewertung sehr subjektiv ist, und sich erfahrungsgemäß zwischen einzelnen Anwendern stark unterscheiden kann, ist der Mittelwert der Antwort über alle Probanden hinweg am aussagekräftigsten. Laut [18] liegt die durchschnittliche Bewertung bei ca. 5 (4,8 – 5,1 ermittelt aus über 200 Tasks bei 5000 Anwendern), was zwar über dem nominellen Mittelpunkt von 4 liegt aber typisch für siebenstufige Likert-Skalen ist. Zu beachten ist hier auch noch, dass bei komplexen Aufgaben die Bewertung generell schlechter ausfällt und es daher notwendig ist, dieselbe Aufgabe mit anderen Systemen bzw. Anwendung zu testen um aussagekräftige Schlüsse ziehen zu können.

3. Hetero-homogenes Data Warehousing

Wie in Kapitel 2.1 beschrieben, ermöglichen Data Warehouses die multidimensionale Analyse von Daten, die aus verschiedenen meist heterogenen Quellen in einem Unternehmen zusammengeführt werden. Diese Heterogenität betrifft beispielsweise die Granularität von Kennzahlen oder deren Maßeinheit. Des Weiteren kann die Heterogenität auch Hierarchien und die darin enthaltenen Ebenen zum Aufrollen oder nichtdimensionale Attribute betreffen. Um Abfragen über diese Daten durchführen zu können, werden diese durch momentane Techniken entweder in ein homogenes Schema zusammengeführt, wodurch wertvolle Informationen verloren gehen können, oder sie werden in ein rein heterogenes Schema überführt, wodurch es zu Problemen bei der Summierbarkeit kommen kann [7]. Durch [2] wurde der hetero-homogene Ansatz eingeführt, der die Nachteile der beiden zuvor genannten Techniken nicht hat. Dieser Ansatz ist dabei so aufgebaut, dass ein minimal homogenes Schema besteht, das Kennzahlen und Hierarchien mit jenen Ebenen enthält, die für alle Daten Gültigkeit besitzen. Für die heterogenen Daten werden Sub-Würfel und Sub-Hierarchien eingeführt. Dabei können die Sub-Hierarchien diese hetero-homogene Eigenschaften besitzen:

- Zusätzliche nichtdimensionale Attribute.
- Zusätzliche Ebenen.

Sub-Würfel können unter anderem diese zusätzlichen Eigenschaften besitzen:

- Zusätzliche Kennzahlen.
- Verschiedene Einheiten für die gleiche Kennzahl (z.B. hat ein Sub-Würfel der sich auf Europa bezieht € als Währung für die Kennzahl „Umsatz“ und ein Sub-Würfel für die USA $ als Währung für „Umsatz“).
- Unterschiedliche Granularität der Kennzahlen (z.B. „Umsatz“ je Stadt im minimalen homogenen Würfel und „Umsatz“ je Geschäft in einem bestimmten Sub-Würfel).
- Verschiedene Kennzahlen unterschiedlicher Granularität in einem Sub-Würfel (z.B. die Kennzahl „Umsatz“ mit der Granularität auf Ebene Geschäft und die Kennzahl „Verkaufte Anzahl“ von Produkten mit der Granularität auf Ebene Stadt).

In diesem Kapitel wird das konzeptuelle, das logische und das physische Modell des hetero-homogenen Ansatzes zur Umsetzung in einem relationalen Datenbanksystem und Mondrian 3 [19] als OLAP Engine beschrieben. Schütz, Neumayr und Schrefl beschreiben in [4], wie hetero-homogene Data Warehouses im Common Warehouse Modell (CWM) modelliert werden können, außerdem wurde von Schütz bereits zuvor ein Prototyp für hetero-homogenes Data Warehousing in Oracle implementiert [3]. Da der Kern dieser Arbeit nicht die eigene und vollständige Implementierung, sondern eine Benutzerstudie ist, werden bestimmte Funktionalitäten des von Schütz implementierten Prototyps verwendet um gewisse Aspekte, die in Mondrian 3 nicht einfach abzudecken sind, abbilden zu können. Die Beispiele und Grafiken zu den Modellen beziehen sich auf das Schema und die Daten die für die Benutzerstudie verwendet werden, allerdings in vereinfachter Form um die Beschreibung nicht unnötig kompliziert zu gestalten. Mit konzeptuellem, logischem und physischem Modell ist die Definition wie im CWM gemeint [20].

Abbildung 9 zeigt ein vereinfachtes Beispiel eines dreidimensionalen Würfel mit den Dimensionen „Customer“ also Kunde, „Store“ also Geschäft und „Time“ also Zeit mit Heterogenität für Geschäfte aus den USA und Kunden aus den USA, die in Geschäften aus den USA eingekauft haben und Geschäften aus Mexico.

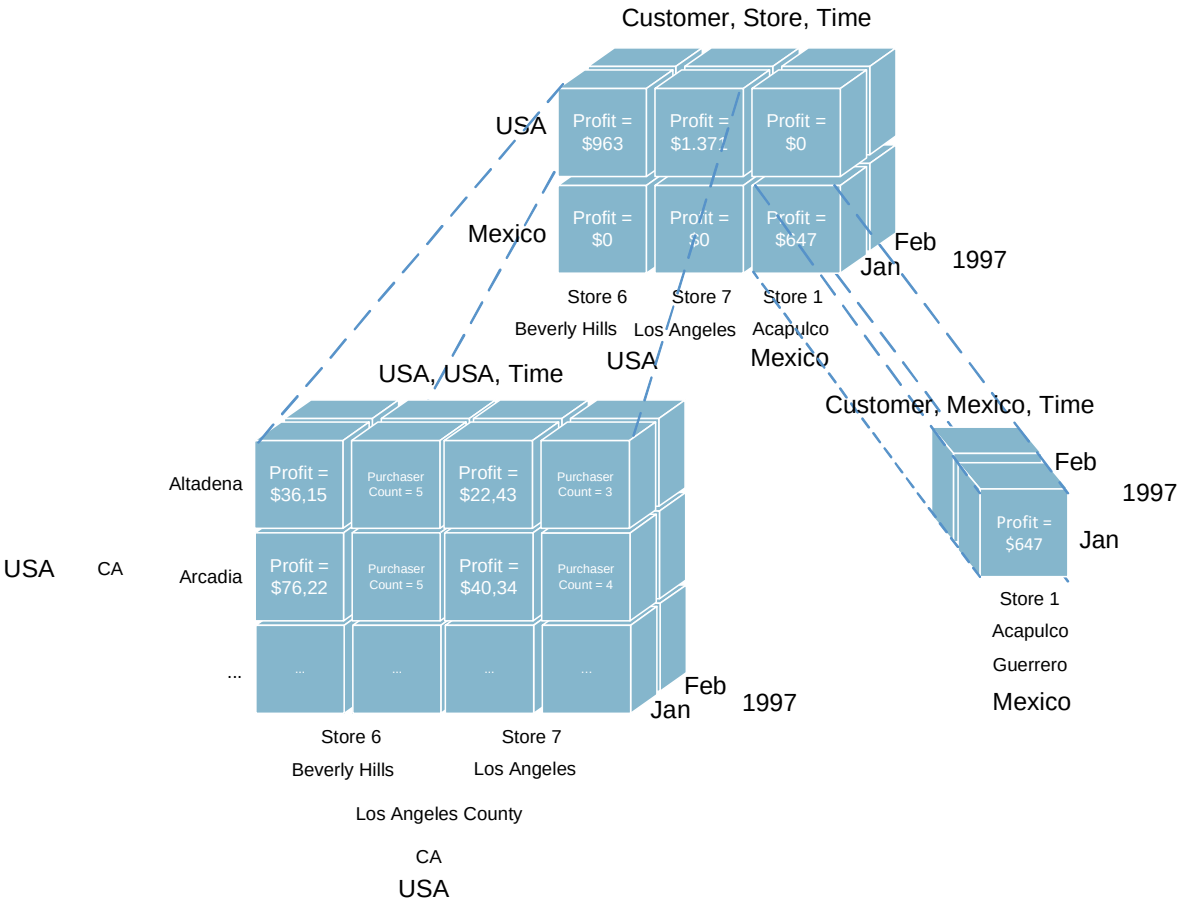


Abbildung 9: Würfel mit Verkaufszahlen und Heterogenität in den USA und Mexico

Im minimal homogenen Schema liegt die Granularität der Verkaufszahlen in der „Customer“ Dimension auf Länderebene, in der „Store“ Dimension auf Geschäftsebene und als Kennzahl ist „Profit“ vorhanden. Für den Bereich im Würfel, in den die Kunden und die Geschäfte aus den USA sind, gibt es eine weitere Kennzahl, nämlich „Purchaser Count“, der die Anzahl der Kunden die in einem Geschäft in einem Monat eingekauft haben repräsentiert. Außerdem stehen für die Dimensionen „Customer“ und „Store“ weitere Ebenen zur Verfügung. Wären nur die Geschäfte aus den USA ausgewählt aber nicht die Kunden, stünden zwar die weiteren Ebenen für diese Dimension zur Verfügung nicht aber die Kennzahl „Purchaser Count“ und weitere Ebenen für Kunden da nicht sichergestellt ist, dass tatsächlich nur Kunden aus den USA in den Geschäften der USA eingekauft haben. Später in den Erklärungen zum logischen und physischen Modell wird das minimal homogene Schema verwendet und die Heterogenität in den USA.

3.1. Konzeptuelles Modell

Mondrian kann selbst nicht entscheiden bei welchen Koordinaten welche Würfelregionen mit welchen Sub-Hierarchien gültig sind, dazu wird um diese semantischen Informationen zu erhalten der von Schütz in Oracle implementierte Prototyp verwendet [3] in dem mehr oder weniger ein Index aufgebaut wird mit dem bestimmbar wird welche Hierarchien und Kennzahlen für welchen Teilbereich im Würfel gelten. Damit kann das konzeptuelle Modell abgebildet werden. Dieser Prototyp verwendet sogenannte M-Obejcts und M-Relationships zur Darstellung von hetero-homogenen Hierarchien und Würfeln in Data Warehouses wie von Neumayr, Schrefl und Thalheim beschrieben [2]. Es werden zunächst M-Objects und anschließend M-Relationships beschrieben.

3.1.1. M-Objects

M-Objects kapseln und ordnen Ebenen in einer partiell linearen Ordnung von der Ebene mit der höchsten Abstraktion bis zu konkretesten. Dabei beschreibt ein M-Object sich selbst und die gemeinsamen Eigenschaften aller Ebenen in der konkretisierten Hierarchie unter sich. So kann eine gemeinsame minimale homogene Hierarchie für eine Dimension gebildet werden in dem ein M-Object als Top-Level definiert wird und speziellere M-Objects all dessen Eigenschaften erben aber heterogene Eigenschaften hinzufügen und so Sub-Hierarchien bilden. Auflistung 1 zeigt wie dies im Prototyp von Schütz implementiert wird, wie zu sehen ist, werden hier die M-Objects angelegt und „USA“ bzw. „Mexico“ erben von dem obersten M-Object „Store“, außerdem ist für jedes M-Object die Eigenschaft hinterlegt welche Mondrian Hierarchie gültig ist. Würde also beispielsweise ein Anwender, in einem OLAP-Tool das diese Basis verwendet, auf der Ebene „Store Country“ einen Filter setzen, um nur Daten für Geschäfte aus den USA zu sehen, würde dieses Tool die Information bekommen, dass hier die Hierarchie „[Store.Store_USA]“ gilt und dementsprechend diese Hierarchie mit ihren weiteren Ebenen verwenden und anzeigen.

```
obj_ref := dim.create_mobject('Store',
                              'all',
                              NULL,
                              level_hierarchy_tty(level_hierarchy_ty('all', NULL),
                                        level_hierarchy_ty('store_country', 'all'),
                                        level_hierarchy_ty('store_city', 'store_country'),
                                        level_hierarchy_ty('store_name', 'store_city')));
utl_ref.select_object(obj_ref, obj);

-- store name of mondrian hierarchy
obj.add_attribute(obj.id || '_mondrian', 'all', 'long_string_tty');
obj.set_attribute(obj.id || '_mondrian',
ANYDATA.convertCollection(long_string_tty('[Store.Store_ALL]')));

obj_ref := dim.create_mobject('USA',
                              'store_country',
                              mobject_trty(dim.get_mobject_ref('Store')),
                              level_hierarchy_tty(level_hierarchy_ty('store_country', 'all'),
                                        level_hierarchy_ty('store_state', 'store_country'),
                                        level_hierarchy_ty('store_county', 'store_state'),
                                        level_hierarchy_ty('store_city', 'store_county'),
                                        level_hierarchy_ty('store_name', 'store_city')));
utl_ref.select_object(obj_ref, obj);

-- store name of mondrian hierarchy
obj.add_attribute(obj.id || '_mondrian', 'store_country', 'long_string_tty');
obj.set_attribute(obj.id || '_mondrian',
ANYDATA.convertCollection(long_string_tty('[Store.Store_USA]')));

obj_ref := dim.create_mobject('Mexico',
                              'store_country',
                              mobject_trty(dim.get_mobject_ref('Store')),
                              level_hierarchy_tty(level_hierarchy_ty('store_country', 'all'),
                                        level_hierarchy_ty('store_entidad', 'store_country'),
                                        level_hierarchy_ty('store_city', 'store_entidad'),
                                        level_hierarchy_ty('store_risk', 'store_city'),
                                        level_hierarchy_ty('store_name', 'store_risk')));
utl_ref.select_object(obj_ref, obj);

-- store name of mondrian hierarchy
obj.add_attribute(obj.id || '_mondrian', 'store_country', 'long_string_tty');
objj.set_attribute(obj.id || '_mondrian',
ANYDATA.convertCollection(long_string_tty('[Store.Store_Mexico]')));
```

Auflistung 1: Hierarchie aus M-Objects im HH-DW Project

3.1.2. M-Relationships

M-Relationships sind analog zu M-Objects und beschreiben die Beziehung zwischen diesen und werden konkretisiert in dem M-Objects aus einer höheren abstrakteren Ebene durch M-Objects die von diesen abgeleitet und verfeinert wurden an bestimmten Koordinaten ausgetauscht werden. Abbildung 10 zeigt wie damit ein hetero-homogener Würfel gebildet wird. Dabei stellen Rechtecke die M-Objects dar und die Raute „Sales_1997“ den Würfel der aus der Verbindung einzelner M-Objects mit ihrer Konkretisierung gebildet wird.

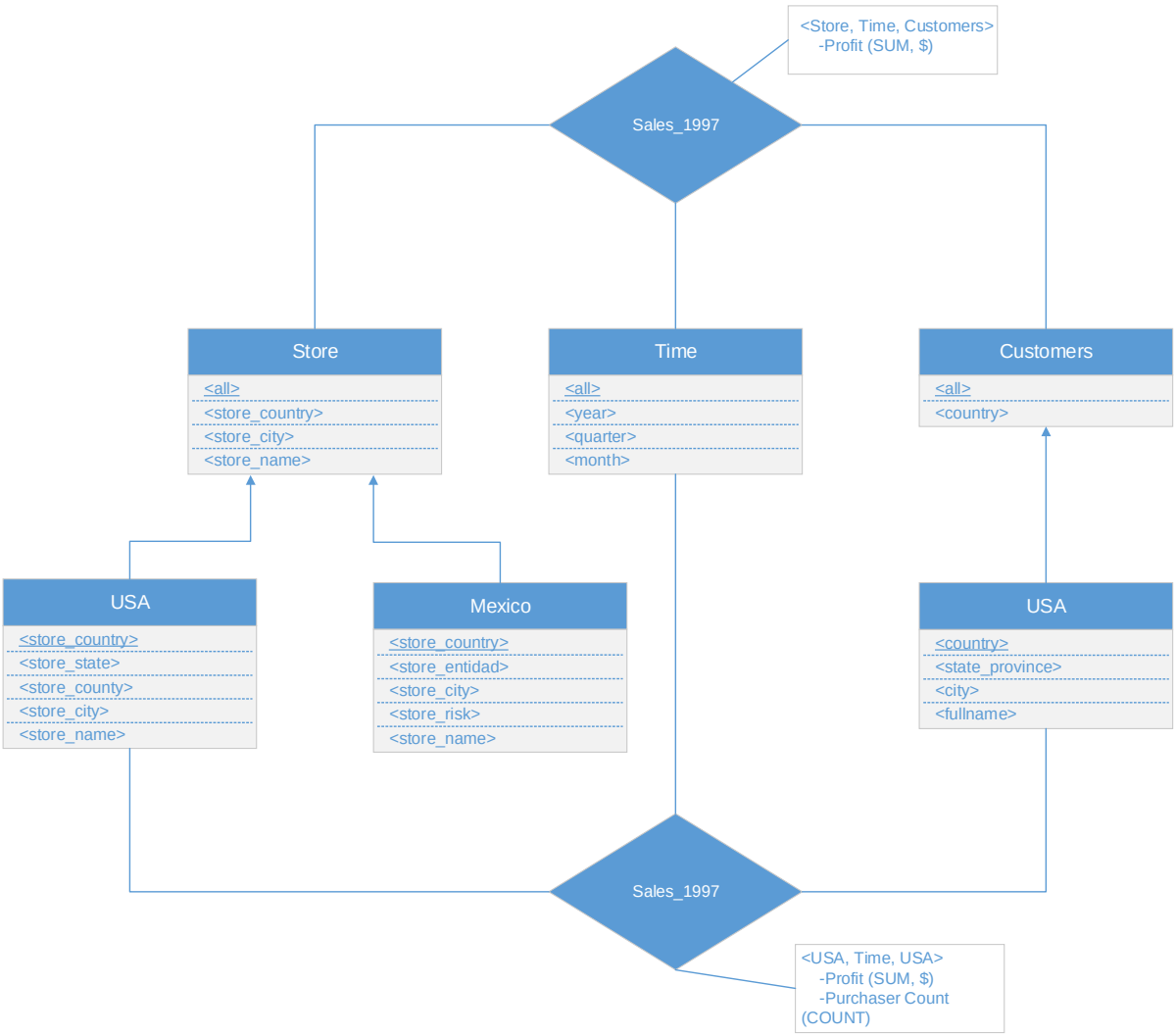


Abbildung 10: M-Relationships die einen hetero-homogenen Würfel bilden

Die M-Relationships bestimmen hier, dass bei den Koordinaten <Store, Time, Customers> die Kennzahl Profit für den Würfel „Sales_1997“ verfügbar ist. Dies entspricht dem minimal homogenen Schema. Wird danach in der Dimension „Stores“ das Land USA ausgewählt, verfügen wir nun zwar über weitere Ebenen, da wir uns jetzt in der Hierarchie für Geschäfte in den USA befinden, aber die Koordinaten <USA, Time, Customers> noch keinen Sub-Würfel mit weiteren Kennzahlen definieren. Werden jetzt auch in der Dimension „Customers“ nur Kunden aus den USA ausgewählt haben wir die Koordinaten <USA, Time, USA> und befinden uns damit in einer Würfelregion, die über eine weitere Kennzahl, nämlich „Purchaser Count“, verfügt.

```
-- Koordinaten für die Würfelregion auf minimal homogener Ebene
mrel_ref := mcube.create_mrel('Store', 'Time', 'Customers');
utl_ref.select_object(mrel_ref, mrel);

-- Profit
mrel.add_measure('profit', conlevel_c000000001_ty('store_name', 'month', 'country'),
'NUMBER');
mrel.set_measure('profit', 'function', false, ANYDATA.convertVarchar2('SUM'));
mrel.set_measure('profit', 'mondrian', true,
ANYDATA.convertVarchar2('[Measures].[Profit_All]'));

-- Koordinate für die Würfelregion mit der Auswahl aller Geschäfte in den USA
mrel_ref := mcube.create_mrel('USA', 'Time', 'Customers');
utl_ref.select_object(mrel_ref, mrel);

-- Profit
mrel.add_measure('profit', conlevel_c000000001_ty('store_name', 'month', 'country'),
'NUMBER');
mrel.set_measure('profit', 'function', false, ANYDATA.convertVarchar2('SUM'));
mrel.set_measure('profit', 'mondrian', true,
ANYDATA.convertVarchar2('[Measures].[Profit_All]'));

-- Koordinate für Würfelregion mit der Auswahl aller Geschäfte in den USA und Kunden aus
den USA
mrel_ref := mcube.create_mrel('USA', 'Time', 'USA');
utl_ref.select_object(mrel_ref, mrel);

-- Purchaser Count
mrel.add_measure('customer_count', conlevel_c000000001_ty('store_name', 'month',
'fullname'), 'NUMBER');
mrel.set_measure('customer_count', 'function', false, ANYDATA.convertVarchar2('COUNT'));
mrel.set_measure('customer_count', 'mondrian', true,
ANYDATA.convertVarchar2('[Measures].[Purchaser Count_USA]'));

-- Profit
mrel.add_measure('profit', conlevel_c000000001_ty('store_name', 'month', 'fullname'),
'NUMBER');
mrel.set_measure('profit', 'function', false, ANYDATA.convertVarchar2('SUM'));
mrel.set_measure('profit', 'mondrian', true,
ANYDATA.convertVarchar2('[Measures].[Profit_USA]'));
```

Auflistung 2: M-Relationships im Hetero-Homogeneous Data Warehouse Project

Auflistung 2 zeigt wie die M-Relations im Hetero-Homogenous Data Warehouse Project von [3] Schütz umgesetzt werden und die entsprechenden Kennzahlen den Koordinaten entsprechend zurückgegeben werden um mit Mondrian genutzt werden zu können. Außerdem ist hier die feinste Granularität der Kennzahl bei den jeweiligen Koordinaten definiert (*mrel.add_measure('profit', conlevel_c000000001_ty(**'store_name', 'month', 'fullname'**), 'NUMBER');*)

3.2. Logisches Modell

Das logische Modell für hetero-homogene Data Warehouses wurde in [4] für das CWM bereits beschrieben, allerdings ist dieses nicht zur Gänze in dieser Form für Mondrian 3 umsetzbar, da im Mondrian-Schema einige Elemente (z.B. CubeRegion das in CWM einen bestimmten Bereich eines Würfels, mit seinen gültigen Ebenen und Kennzahlen, beschreibt) nicht vorhanden bzw. in anderer Form umgesetzt sind und prinzipiell in einem Mondrian Würfel nur eine Faktentabelle sein kann und außerdem das Mapping der Dimensionen bzw. der einzelnen Ebenen auf das physische Modell nicht ident funktioniert. So können einige wichtige Aspekte nicht abgebildet werden und benötigen ein Workaround. Die Modellierung des hetero-homogenen Data Warehouse in Mondrian wurde dennoch soweit möglich aus dem zuvor angesprochenen Paper abgeleitet.

Das Mondrian-Schema wird in einem XML-Dokument definiert und beinhaltet das logische Modell und das Mapping auf die physische Ebene. Die wichtigsten Komponenten in einem Mondrian-Schema sind <Dimension>, wie der Name bereits sagt eine Dimension, <Cube>, ein Würfel, und <Measure> eine Kennzahl. Außerdem wichtig für den hetero-homogenen Ansatz ist <VirtualCube> eine Würfel aus mehreren anderen Würfeln [19]:

- <Dimension> stellt den Kontext für die Analyse dar (beispielsweise Zeit, Ort, Produkt, usw.) und kann eine oder mehrere Hierarchien (beispielsweise kann die Dimension Zeit eine Hierarchie mit den Ebenen Jahr, Monat, Tag und eine Hierarchie mit den Ebenen Jahr, Quartal, Woche enthalten) mit einer oder mehreren Ebenen enthalten über die aufgerollt werden kann und damit die Granularität einer Kennzahl bestimmen.
- <Cube> ist die Sammlung aus einer oder mehrerer Dimensionen und Kennzahlen und stellt damit das wichtigste Element dar, da in ihm alle interessanten- bzw. wichtigen Daten gehalten und organisiert werden.
- <Measure> ist eine Größe von Interesse die beispielsweise die Kosten oder den Ertrag für ein Produkt im Data Warehouse repräsentiert.
- <VirtualCube> erlaubt das Zusammenfügen mehrerer Würfel und die Nutzung von gemeinsamen Dimensionen („Shared Dimensions“) für Abfragen über diese Würfel.

Zuerst wird gezeigt wie Dimensionen und ihre Hierarchien in Mondrian modelliert werden um für den hetero-homogenen Ansatz genutzt werden zu können, danach Würfel und wie diese und gemeinsame Dimensionen in einem <VirtualCube> zusammengeführt werden.

3.2.1. Dimensionen

Wie bereits in Kapitel 2.1.1 zu multidimensionaler Modellierung dargelegt, bestehen Dimensionen aus einer Vielzahl von Elementen, die einer bestimmten Ebene in einer Hierarchie zugeordnet sind. Weitere Elemente, in der nächst feiner granularen Ebene einer Hierarchie, können genau einem Element aus der gröberen Ebene zugeordnet sein und werden aggregiert. Da Dimensionen aber in ihrer Gesamtheit heterogen sein können, ist es notwendig mehrere Hierarchien in der Dimension zu bilden, die für sich homogen sind um der Definition des hetero-homogenen Ansatzes zu entsprechen, dies kann rekursiv auf gebildete Hierarchien angewendet werden um kleinere Bereiche noch weiter zu verfeinern. Sieht man sich Abbildung 9 an sind Mexico und die USA in Bezug auf ihre Ebenen und eine Kennzahl die nur in Geschäften aus den USA bei Kunden aus den USA vorhanden sind, heterogen. Sie weisen aber Ebenen auf die sich beide teilen. Dies wäre beispielsweise für Geschäfte die Ebene mit der feinsten Granularität „Store Name“ mit den Elementen Store 6, Store 7 und Store 1, die Ebene „Store City“ mit den Elementen Beverly Hills, Los Angeles und Acapulco und die Ebene „Store Country“ mit den Elementen USA und Mexico.

```
<Dimension name="Store">
    <Hierarchy name="Store_ALL" hasAll="true"
               primaryKey="store_name">
        <Table name="stores_view_mhv"/>
        <Level name="Store Country" column="store_country"/>
        <Level name="Store City" column="store_city"/>
        <Level name="Store Name" column="store_name"/>
    </Hierarchy>
    <Hierarchy name="Store_USA" hasAll="true"
               primaryKey="store_name">
        <Table name="stores_view_usa"/>
        <Level name="Store Country" column="store_country"/>
        <Level name="Store State A" column="store_state"/>
        <Level name="Store County A" column="store_county"/>
        <Level name="Store City" column="store_city"/>
        <Level name="Store Name" column="store_name"/>
    </Hierarchy>
    <Hierarchy name="Store_Mexico" hasAll="true"
               primaryKey="store_name">
        <Table name="stores_view_mexico"/>
        <Level name="Store Country" column="store_country"/>
        <Level name="Store County C" column="store_entidad"/>
        <Level name="Store City" column="store_city"/>
        <Level name="Store Risk" column="store_risk"/>
        <Level name="Store Name" column="store_name"/>
    </Hierarchy>
</Dimension>
```

Auflistung 3: Dimension Store in Mondrian-Schema

In Mondrian lässt sich dies so lösen, dass in diesem konkreten Beispiel drei Hierarchien für die Dimension „Store“ gebildet werden. Einerseits eine Hierarchie die nur Ebenen enthält, die sowohl für die Geschäfte in den USA als auch die Geschäfte in Mexico gelten, also eine minimal homogene Hierarchie und je eine Hierarchie für Geschäfte in den USA und für Geschäfte in Mexiko. Auflistung 3 zeigt diese drei Hierarchien. Sowohl Mexico als auch die USA behalten in den Hierarchien, die nur für sie selbst gelten, die Informationen der Hierarchie die für beide gemeinsam gültig sind, haben aber zusätzlich Informationen die nur in ihrem Bereich gültig sind. Sie wären also im gesamten heterogen, bleiben aber in ihrem eigenen Gültigkeitsbereich homogen. Die XML-Elemente <Table> in jeder Hierarchie und „column“ Attribute dienen dem Mapping auf die physischen Tabellen und ihre Spalten und werden später im Kapitel zur physischen Modellierung näher beschrieben.

3.2.2. Würfel

In [4], das die Modellierung des hetero-homogenen Ansatzes in CWM beschreibt, ist ein „Cube“ ein Konstrukt, das keine Ebene der Granularität für eine Kennzahl garantiert oder bestimmt. Vielmehr beschreibt dieser was zumindest von den Daten im Würfel erwartet werden kann. Um bestimmte Teilbereiche im Würfel repräsentieren zu können werden Würfelregionen „CubeRegions“, verwendet. Diese repräsentieren für sich, bezogen auf in diesem Teilbereich enthaltene Kennzahlen, ein homogenes Schema und gelten auch nur für diesen. Dies funktioniert also gleich wie bei den Dimensionen, eine „CubeRegion“ für sich ist homogen aber bezogen auf ein globales Schema heterogen. In Mondrian kann dies durch den <VirtualCube> umgesetzt werden. Dieser kann mehre Würfel, die dann „CubeRegions“ repräsentieren, und Dimensionen vereinen. Die Bedingung hier ist aber, dass diese Dimensionen auch im ganzen Schema gültig sind, sogenannte „Shared Dimensions“, allerdings können Würfel nur auf Hierarchien in diesen Dimensionen aufrollen, wenn sie im Mapping auf der physischen Ebene, also der Faktentabelle, einen Dimensions Key haben, der sie mit der Dimensionstabelle verbindet.

Das Schema in Auflistung 4 zeigt ein hetero-homogenes Schema in Mondrian, der <VirtualCube> fasst die beiden Würfel „Sales_1997_USA“ und „sales_1997_MHV“ durch <CubeUsage> zusammen und definiert durch die Elemente <VirtualCubeDimension> und <VirtualCubeMeasure> welche Dimensionen und Kennzahlen im virtuellen Würfel verwendet werden.

```
<?xml version="1.0"?>
<Schema name="FoodMartTestHH">
    <!-- Shared dimensions -->
    <Dimension name="Store">
        …
    </Dimension>
    <Dimension name="Customers">
        <Hierarchy name="Customers_ALL" hasAll="true" allMemberName="All Customers"
                   primaryKey="country">
            <Table name="customers_view_mhv"/>
            <Level name="Customer Country" column="country"/>
        </Hierarchy>
        <Hierarchy name="Customers_USA" hasAll="true" primaryKey="fullname">
            <Table name="customers_view_usa"/>
            <Level name="Customer Country" column="country"/>
            <Level name="Customer State Province" column="state_province"/>
            <Level name="Customer City" column="city"/>
            <Level name="Customer Name" column="fullname"/>
        </Hierarchy>
    </Dimension>
    <Cube name="Sales_1997_MHV" visible="false">
        <Table name="sales_facts_1997_mhv"/>
        <DimensionUsage name="Store" source="Store" foreignKey="store"/>
        <DimensionUsage name="Time" source="Time" foreignKey="time"/>
        <DimensionUsage name="Customers" source="Customers" foreignKey="customer"/>
        <Measure name="Profit_All" column="profit" aggregator="sum"/>
    <Cube name="Sales_1997_USA" visible="false">
        <Table name="sales_facts_1997_usa"/>
        <DimensionUsage name="Store" source="Store" foreignKey="store"/>
        <DimensionUsage name="Time" source="Time" foreignKey="time"/>
        <DimensionUsage name="Customers" source="Customers" foreignKey="customer"/>
        <Measure name="Purchaser Count_USA" column="customer" aggregator="distinct-count"/>
        <Measure name="Profit_USA" column="profit" aggregator="sum"
                 formatString="Standard"/>
    </Cube>
    <VirtualCube name="Sales_1997">
        <CubeUsages>
            <CubeUsage cubeName="Sales_1997_MHV"/>
            <CubeUsage cubeName="Sales_1997_USA"/>
        </CubeUsages>
        <VirtualCubeDimension name="Store" source="Store" foreignKey="store"/>
        <VirtualCubeDimension name="Customers" source="Customers" foreignKey="customer"/>

        <VirtualCubeMeasure cubeName="Sales_1997_MHV" name="[Measures].[Profit_All]"/>

        <VirtualCubeMeasure cubeName="Sales_1997_USA"
                            name="[Measures].[Purchaser Count_USA]"
                            aggregator="distinct-count"/>
        <VirtualCubeMeasure cubeName="Sales_1997_USA" name="[Measures].[Profit_USA]"/>
    </VirtualCube>
</Schema>
```

Auflistung 4: Mondrian-Schema mit <VirtualCube>

3.3. Physisches Modell

In diesem Kapitel wird erklärt wie aus den in der relationalen Datenbank gehaltenen heterogenen Dimensionstabellen und Faktentabellen hetero-homogene dem logischen Modell entsprechende Tabellen gebildet werden und wie das Mapping vom physischen auf das logische Modell in Mondrian funktioniert.

3.3.1. Dimensionstabellen

Dimensionstabellen in relationalen Data Warehouses enthalten üblicherweise für jede Ebene zu einer Hierarchie im logischen Modell eine Spalte und entweder eine weitere Spalte mit einem Schlüssel über den sie mit der Faktentabelle verbunden werden kann, um ein Star-Schema zu bilden, oder sie wird über die Spalte welche die kleinste Ebene darstellt mit der Faktentabelle verknüpft. Tabelle 1 zeigt die heterogene Dimension „Store“ aus den vorigen Beispielen.

store_country	store_state	store_entidad	store_county	store_city	store_risk	store_name
Mexico	NULL	Guerrero	NULL	Acapulco	green	Store 1
USA	CA	NULL	Los Angeles County	Beverly Hills	NULL	Store 6
USA	CA	NULL	Los Angeles County	Los Ange-les	NULL	Store 7

Tabelle 1: Heterogene Dimensionstabelle „Store"

Wie zu sehen ist, gibt es in dieser Tabelle Heterogenität in den Spalten store_state, store_entidad und store_risk. Um diese Dimension sauber für den hetero-homogenen Ansatz in Mondrian verwendbar machen zu können müssen drei materialisierte Views erzeugt werden, die jene Hierarchien aus dem logischen Schema repräsentieren, die in Auflistung 3 gezeigt wurden. Auflistung 5 zeigt wie die Tabelle mit einer minimal homogenen Hierarchie für die Dimension „Stores“, die Tabelle für eine homogene Hierarchie für Geschäfte in Mexico und die Tabelle für eine homogene Hierarchie für Geschäfte in den USA in MySQL gebildet wird.

```
CREATE OR REPLACE VIEW stores_view_mhv
AS SELECT store_country, store_city, store_name
FROM store;

CREATE OR REPLACE VIEW stores_view_mexico
AS SELECT store_country, store_entidad, store_city, store_risk, store_name
FROM store
WHERE store_country = 'Mexico';

CREATE OR REPLACE VIEW stores_view_usa
AS SELECT store_country, store_state, store_county, store_city, store_name
FROM store
WHERE store_country = 'USA';
```

Auflistung 5: Queries zur Erzeugung der Hierarchien für die "Store" Dimension

Tabelle 2 zeigt das Ergebnis der dritten Query um einen View für die homogene Hierarchie der Geschäfte in den USA zu erstellen.

store_country	store_state	store_county	store_city	store_name
USA	CA	Los Angeles County	Beverly Hills	Store 6
USA	CA	Los Angeles County	Los Angeles	Store 7

Tabelle 2: Homogener USA View für die Dimension "Store"

Um diese Tabelle mit der Hierarchie für Stores in den USA mit dem logischen Modell zu verbinden, muss im Mondrian-Schema in dieser Hierarchie der Schlüssel und die Tabelle angegeben werden. Auflistung 6 zeigt die Hierarchie „Store_USA“ die sich durch primaryKey=“store_name“ mit der Tabelle „stores_view_usa“ verbindet und über die <Level> Elemente bestimmt wie in dieser Hierarchie aufgerollt werden muss indem jeder <Level> einer Spalte in der Tabelle zugewiesen wird. So ist in dieser Hierarchie „Store Country“ die gröbste granulare Ebene und „Store Name“ die feinst granulare.

```
<Dimension name="Store">
    …
    <Hierarchy name="Store_USA" hasAll="true" primaryKey="store_name">
        <Table name="stores_view_usa"/>
        <Level name="Store Country" column="store_country"/>
        <Level name="Store State A" column="store_state"/>
        <Level name="Store County A" column="store_county"/>
        <Level name="Store City" column="store_city"/>
        <Level name="Store Name" column="store_name"/>
    </Hierarchy>
    …
</Dimension>
```

Auflistung 6: Mondrian Hierarchie für Geschäfte in den USA

3.3.2. Faktentabellen

Eine Faktentabelle hat üblicherweise eine Spalte für jede Dimension und jede Kennzahl, außer diese Kennzahl berechnet sich beispielsweise wie die Kennzahl „Purchaser Count“ aus anderen Informationen in der Faktentabelle („Purchaser Count“ ermittelt die Anzahl der Kunden indem sie jeden eindeutigen Kunden zählt). Um in Mondrian den hetero-homogenen Ansatz abbilden zu können wird, wie bereits erwähnt der <VirtualCube> verwendet, der mehrere Würfel zusammenschließt und so einen heterogenen Würfel darstellt. Damit dies genutzt werden kann, müssen aus der heterogenen Faktentabelle so viele für sich homogene Views geschaffen werden wie es heterogene Kennzahlen gibt, die nicht in einer homogenen Faktentabelle gehalten werden können. Im Fall des hier verwendeten Beispiels müssen physisch also zwei Views, die jeweils für sich homogen aber im Ganzen betrachtet heterogen sind, gebildet werden. Einerseits ist das eine minimal homogene Sicht und andererseits eine Sicht auf die Region des Würfels der Geschäfte aus den USA in denen Kunden aus den USA Profit erzeugt haben. Tabelle 3 zeigt einen Ausschnitt der heterogenen Faktentabelle.

store_id	time_id	customer_id	profit
1	469	1	1,2568
1	493	1	3,0324
1	501	1	1,4962
6	372	172	4,5942
6	402	5389	1,7172
7	385	527	4,347
7	378	551	1,464

Tabelle 3: Heterogene Faktentabelle „sales_facts_1997“

Diese Tabelle ist heterogen, da in Mexico Kunden nicht identifiziert werden können, und darum bei jedem Einkauf jeder Kunden die gleiche customer_id bekommt, in diesem Fall die Nummer 1. Kunden aus den USA haben eine Kundenkarte die bei jedem Einkauf an Geschäften in den USA an der Kasse eingelesen wird und diese so eindeutig identifiziert. Damit kann ermittelt werden wie viele einzelne Kunden eingekauft haben. Durch die Kundenkarte sind weitere Informationen wie Wohnort usw. in der Dimensionstabelle vorhanden. Dadurch ist auch eine feinere Granularität in der Ermittlung des Profits möglich, da so sichtbar ist aus welchen Gebieten welche Kunden wo einkaufen und nicht nur mit der Granularität des gesamten Landes, wie es in der Würfelregion für Kunden aus Mexico der Fall ist. Um nun eine Faktentabelle für die minimal homogene Sicht zu schaffen ist die Query aus Auflistung 7 nötig. In diesem View werden die IDs der Dimensionen durch die Ebene mit der feinsten Granularität für die jeweilige Dimension die in dieser Sicht verfügbar ist ersetzt. Da, wie bereits erwähnt einzelne Kunden in Mexico nicht identifiziert werden können, wird um Heterogenität zu eliminieren, die feinste Granularität auf die Ebene des Landes beschränkt.

```
CREATE OR REPLACE VIEW sales_facts_1997_mhv
AS SELECT s.store_name AS store, t.time_id AS time, c.country AS customer,
SUM(sf.profit) AS profit
FROM stores_hh s, time_hh t, customers_hh c, sales_facts_1997 sf
WHERE s.store_id = sf.store_id
      AND t.time_id = sf.time_id
      AND c.customer_id = sf.customer_id
GROUP BY store, time, customer;
```

Auflistung 7: Sicht minimal homogener Sales Würfel

Für die Würfelregion mit den Koordinaten <USA, Time, USA>, wie auch in Abbildung 10 gezeigt, wird der View durch die Query aus Auflistung 8 gebildet, in dem nur Geschäfte aus den USA und Kunden aus den USA enthalten sind, aber mit einer deutlich feineren Granularität in der Kundendimension, die bis auf den Namen hinunter geht, und einer weiteren Kennzahl „Purchaser Count“, die sich durch das zählen einzelner Kunden ermitteln lässt, besitzt.

```
CREATE OR REPLACE VIEW sales_facts_1997_usa
AS SELECT s.store_name AS store, t.time_id AS time, c.fullname AS customer,
SUM(sf.profit) AS profit
FROM stores_hh s, time_hh t, customers_hh c, sales_facts_1997 sf
WHERE s.store_id = sf.store_id
      AND t.time_id = sf.time_id
      AND c.customer_id = sf.customer_id
      AND s.store_country = 'USA'
      AND c.country = ‘USA’
GROUP BY product, time, customer, promotion, store;
```

Auflistung 8: Sicht auf Würfel mit Geschäften und Kunden aus den USA

Wie das Mapping von physischen Hierarchien mit dem logischen Schema von Mondrian funktioniert wurde bereits erörtert, wie das Mapping auf diese Würfel funktioniert zeigt Auflistung 9 die einen Ausschnitt eines Mondrian-Schemas darstellt.

```
<Dimension name="Customers">
    <Hierarchy name="Customers_ALL" hasAll="true" allMemberName="All Customers"
        primaryKey="country">
        <Table name="customers_view_mhv"/>
        <Level name="Customer Country" column="country"/>
    </Hierarchy>
    <Hierarchy name="Customers_USA" hasAll="true" primaryKey="fullname">
        <Table name="customers_view_usa"/>
        <Level name="Customer Country" column="country"/>
        <Level name="Customer State Province" column="state_province"/>
        <Level name="Customer City" column="city"/>
        <Level name="Customer Name" column="fullname"/>
    </Hierarchy>
</Dimension>
<Cube name="Sales_1997_MHV" visible="false">
    <Table name="sales_facts_1997_mhv"/>
    …
    <DimensionUsage name="Customers" source="Customers" foreignKey="customer"/>
    <Measure name="Profit_All" column="profit" aggregator="sum"/>
<Cube name="Sales_1997_USA" visible="false">
    <Table name="sales_facts_1997_usa"/>
    …
    <DimensionUsage name="Customers" source="Customers" foreignKey="customer"/>
    <Measure name="Purchaser Count_USA" column="customer" aggregator="distinct-count"/>
    <Measure name="Profit_USA" column="profit" aggregator="sum"
    formatString="Standard"/>
</Cube>
<VirtualCube name="Sales_1997">
    <CubeUsages>
        <CubeUsage cubeName="Sales_1997_MHV"/>
        <CubeUsage cubeName="Sales_1997_USA"/>
    </CubeUsages>
    …
    <VirtualCubeDimension name="Customers" source="Customers" foreignKey="customer"/>

    <VirtualCubeMeasure cubeName="Sales_1997_MHV" name="[Measures].[Profit_All]"/>

    <VirtualCubeMeasure cubeName="Sales_1997_USA"  name="[Measures].[Purchaser
    Count_USA]" aggregator="distinct-count"/>
    <VirtualCubeMeasure cubeName="Sales_1997_USA"name="[Measures].[Profit_USA]"/>
</VirtualCube>
```

Auflistung 9: Mondrian-Schema Ausschnitt mit Customers Dimension

Der Ausschnitt zeigt die Dimension „Customers“ mit ihrer minimal homogenen Hierarchie, die nur die Ebene „Customer Country“ besitzt und die zweite Hierarchie, die mehrere Ebenen bis zur feinsten Granularität auf Namensebene eines einzelnen Kunden. Diese Dimension ist, wie alle anderen bei der Umsetzung des hetero-homogenen Ansatzes in Mondrian, außerhalb der <Cube> Elemente im Schema angeordnet, da sie nur so von allen verwendet werden können. Durch das Element <DimensionUsage> innerhalb eines Würfels kann auf die Dimension außerhalb dieses Würfels zugegriffen werden, dabei muss ein Fremdschlüssel angegeben werden, der auf die Faktentabelle des Würfels zeigt und der Primärschlüssel in der Dimension ist jener der gerade verwendeten Hierarchie in dieser Dimension. Dementsprechend kann der minimal homogene Würfel „Sales_1997_MHV“ nichts mit der

Hierarchie „Customers_USA“ in der „Customers“ Dimension anfangen, da der Fremdschlüssel in seiner Faktentabelle in der „customer“ Spalte keine Übereinstimmung mit dem Primärschlüssel in der „Customers_USA“ Hierarchie die dort der Kundenname ist finden wird, diesen hat aber die Faktentabelle des „Sales_1997_USA“ Würfels. An diesem Punkt kommt der <VirtualCube> ins Spiel, dieses Konstrukt erlaubt es, die beiden anderen Würfel durch das Element <CubeUsage> zusammenzuführen, außerdem kann er durch <VirtualCubeDimension> und <VirtualCubeMeasure> auch alle vorhandenen Dimensionen und Kennzahlen aus den Würfeln die er umschließt verwenden. Damit kann das beschriebene Konzept zur Umsetzung in CWM wie es in [4] beschrieben ist, zum Teil umgesetzt werden. Es muss allerdings der Prototyp von Schütz [3], wie weiter oben beschrieben, verwendet werden um feststellen zu können in welcher Würfelregion sich ein Anwender befindet und einen Wechsel von Hierarchien und Kennzahlen auf die in diesem Bereich verfügbaren für den Anwender transparent zu ermöglichen.

4. Hetero-homogenes OLAP

In diesem Kapitel wird das Konzept zur Umsetzung des hetero-homogenen Ansatzes für das webbasierte OLAP Frontend Saiku 2.4 und die prototypische Implementierung dazu beschrieben.

4.1. Konzept

Wie in Kapitel 3 beschrieben, ist es mit einem Mondrian-Schema zwar möglich die Grundstrukturen aufzubauen um den hetero-homogenen Ansatz in einem relationalen Data Warehouse abzubilden, jedoch mangelt es an der Fähigkeit von Mondrian durch einen Würfel mit heterogenen Informationen zu navigieren, wie es der hetero-homogene Ansatz verlangt. Es wären für den Anwender, neben der minimal homogenen Hierarchie, auch die jeweiligen Sub-Hierarchien der Dimension und heterogenen Kennzahlen sichtbar und der Anwender müsste selbst entscheiden, welche Hierarchie und welche Kennzahl den jeweiligen Würfelbereich abbildet. Mondrian fährt hier bildlich gesprochen blind durch den Raum. Aus diesem Grund ist es nötig Mondrian, bzw. dem Frontend des verwendeten OLAP-Tools, eine Möglichkeit zu geben entscheiden zu können welche Hierarchien und Kennzahlen es einem Anwender auf Grund der Slice und Dice Operationen die dieser durchführt, zeigen muss und welche es zu verbergen hat wenn sich der Anwender auf Grund seiner Suchoperationen in einem Bereich des Würfels befindet, in dem diese nicht vorhanden sind. Dieser ganze Vorgang muss dabei transparent vom Anwender erfolgen. Ein Weg dies zu ermöglichen, ohne die Mondrian OLAP Engine selbst zu verändern, ist einen Dienst zwischenzuschalten, der immer sobald der Anwender Slice und Dice Operationen durchführt, diese Suchparameter prüft indem er den bestehenden hetero-homogenen Prototyp verwendet, der in Kapitel 3.1 beschrieben wurde, und die gültigen Hierarchien und Kennzahlen ermittelt und zurückgibt. Sind diese gleich wie zuvor, braucht das Frontend nichts zu verändern. Sind diese aber verschieden, muss das Frontend die neuen Hierarchien und Kennzahlen zeigen und die alten ausblenden. Außerdem muss es die bisher vom Anwender bereits ausgewählten Ebenen und Kennzahlen in den neuen Hierarchien durch dieselben ersetzen und wieder die richtigen Filter setzen. Das Aktivitätsdiagramm in Abbildung 11 zeigt diesen Vorgang.

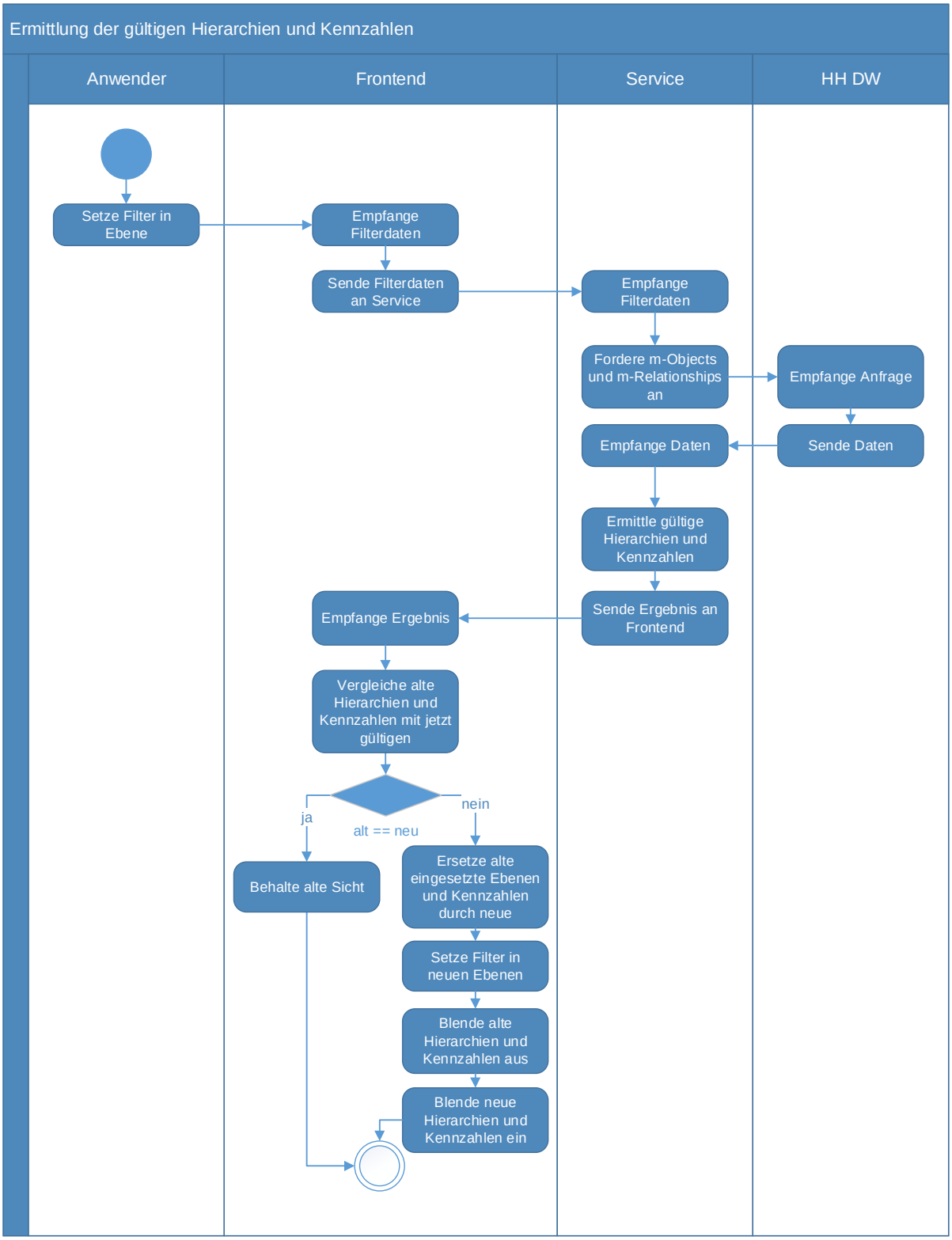


Abbildung 11: Aktivitätsdiagramm hetero-homogenes Frontend

4.2. Implementierung

Bei der Anwendung in der das zuvor beschrieben Konzept umgesetzt wird, handelt es sich um Saiku 2.4 [12], welche in Kapitel 2.2 vorgestellt wurde. Das Frontend von Saiku welches verändert werden muss, um die angesprochene Funktionalität zu gewährleisten wurde in JavaScript implementiert. Es verwendet für einen strukturierten Aufbau und um Daten zwischen Client und Server synchron zu halten Backbone.js [21], eine JavaScript-Bibliothek welche eine MV* Design-Pattern darstellt und von der JavaScript-Bibliothek Underscore.js [22] abhängig ist, die eine Sammlung von Werkzeugen für Operationen auf Container-Objekte wie Arrays usw. und JavaScript-Funktionen ohne Objekt-Prototypen zu verwenden zur Verfügung stellt. Des weiteren verwendet Saiku jQuery [23] eine weit verbreitet und sehr beliebte JavaScript Bibliothek die zur Navigation und Veränderung des DOM (Document Object Model) einer Webseite verwendet wird.

Zunächst wird in Kapitel 4.2.1 überblickhaft der Aufbau des Frontends bzw. der JavaScript Dateien gegeben, anschließend werden in Kapitel 4.2.2 die Änderungen und das Hinzufügen von eigenen Funktionalitäten beschrieben.

4.2.1. Saiku - Aufbau

Abbildung 12 zeigt den strukturellen Aufbau des Saiku 2.4 Frontends. Index.html ist die Seite, die dem Anwender in einem beliebigen Browser als Singlepageanwendung dargestellt wird.

In der index.html Datei werden alle in Abbildung 12 aufgelisteten Skriptdateien und weitere wie jQuery.js, Backbone.js, Underscore.js und diverse Plug-Ins, bei Start geladen. Saiku.js ist das zentrale Objekt zum Handling des globalen Zustands der Anwendung. Es hält also hierarchisch betrachtet alle Models, Views und Events. Models repräsentieren Daten, die über die einzelnen Views dargestellt werden. Diese Daten beziehen die Models bei ihrer Initialisierung über SaikuServer.js der AJAX-Calls setzt und so mit dem Webserver kommuniziert. Außerdem beziehen so die Models neue Daten vom Webserver bei Aktionen des Anwenders die Änderungen erfordern die nicht alleine auf Clientseite verfügbar sind. Diese Aktionen bzw. Events sind an Views gebunden, die durch den Anwender ausgelöst werden können und auf Grund dessen sich Models und Views wieder ändern können. In der index.html Datei sind einzelne Templates die bestimmten Models bzw. Views zugeteilt sind und so letztendlich die Seite aufbauen.

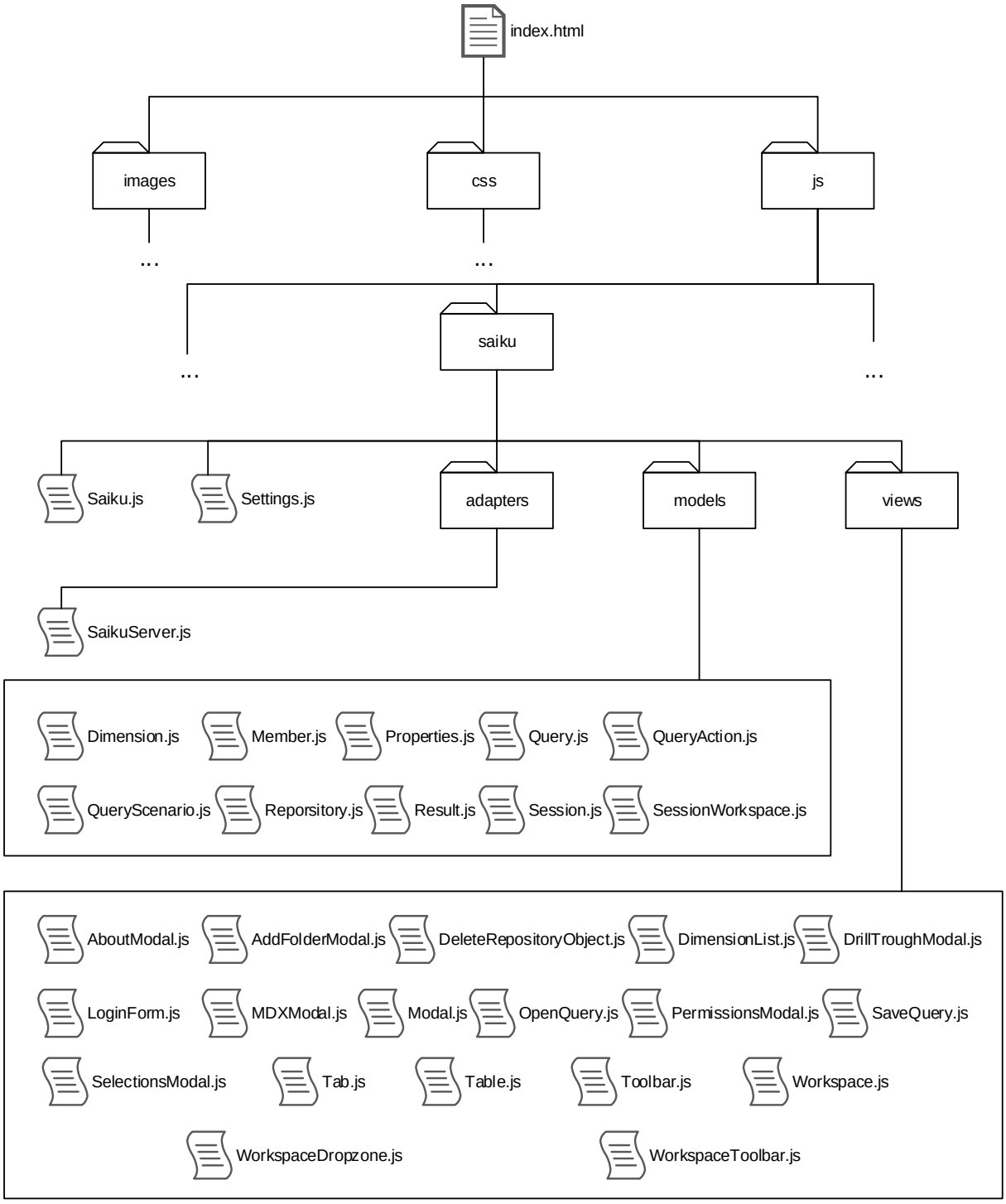


Abbildung 12: Saiku Frontend Struktur

4.2.2. Saiku - Änderungen

In diesem Kapitel werden die Änderungen am Frontend von Saiku 2.4 beschrieben. Zuerst werden notwendige Namenskonventionen für die Hierarchien und Kennzahlen gegeben, anschließend werden die Änderungen in der Settings.js und index.html Datei beschrieben, danach die restlichen Änderungen an den Models und Views und zum Abschluss die Implementierung des Scripts HeteroHomogenousRuleset.js um die richtigen Daten für den hetero-homogenen Ansatz zur Verfügung zu stellen.

Namenskonventionen

Da wie in Kapitel 3 beschrieben in einer Dimension eine minimal homogene Hierarchie und mehrere Sub-Hierarchien und außerdem alternative Hierarchien sein können und diese jeweils einen eindeutigen Namen benötigen, der die ID darstellt und dies für den Anwender nicht sichtbar sein soll ist es notwendig Namenskonventionen einzuführen um zu Gewährleisten, dass die Hierarchien die ausgetauscht werden müssen intern erkannt und zugeordnet werden können und gleichzeitig für den Anwender nicht sichtbar ist, dass eine andere Hierarchie angezeigt wird. Dazu wir ein Unterstrich „_“ verwendet, der erste Teil der Hierarchie trägt die Zeichenfolge die für den Anwender sichtbar sein soll und der Teil nach dem Unterstrich für die minimal homogenen Sichten „ALL“ bzw. für Sub-Hierarchien eine beliebige, aber aus Verständlichkeitsgründen sinnvolle, Zeichenfolge. Wichtig ist, sollte in einer Dimension eine alternative Hierarchie auswählbar sein und diese ist selbst eine minimal homogene, so muss zumindest der erste Buchstabe dieser Hierarchie ein anderes Zeichen enthalten als die andere verfügbare Hierarchie und ihre Sub-Hierarchien. Auflistung 10 zeigt wie diese Konvention für die Hierarchien der Dimension „Product“ umgesetzt wird. Hier hat „Product by Department“ insgesamt vier Hierarchien, nämlich die minimal homogene „Product by Department_ALL“ und die drei Sub-Hierarchien „Product by Department_Multicategory“, „Product by Department_Subcategorized“ und „Product by Department_FreshFruit“ und „Branded“ als alternative Hierarchie besteht aus der minimalen Hierarchie mit „Branded_ALL“ und der Sub-Hierarchie mit „Branded_Product“.

```
<Dimension name="Product">
        <Hierarchy name="Product by Department_ALL" hasAll="true"
        primaryKey="product_name">
        …
        </Hierarchy>
        <Hierarchy name="Product by Department_Multicategory" hasAll="true"
        primaryKey="product_name">
        …
        </Hierarchy>
        <Hierarchy name="Product by Department_Subcategorized" hasAll="true"
        primaryKey="product_name">
        …
        </Hierarchy>
        <Hierarchy name="Product by Department_FreshFruit" hasAll="true"
        primaryKey="product_name">
        …
        </Hierarchy>
        <Hierarchy name="Branded_ALL" hasAll="true" primaryKey="product_name">
        …
        </Hierarchy>
        <Hierarchy name="Branded_Product" hasAll="true" primaryKey="product_name">
        …
        </Hierarchy>
</Dimension>
```

Auflistung 10: Hierarchien-Namen Dimension „Product"

Für den Anwender ist nach der entsprechenden Änderung des Templates in der index.html dann nur die Zeichenfolge vor dem Unterstrich sichtbar. Die gleiche Konvention gilt für Kennzahlen und wird in Auflistung 11 dargestellt.

```
        <Measure name="Store Sales_All" column="store_sales" aggregator="sum"
        formatString="$#,###.00"/>
        <Measure name="Store Sales_USA" column="store_sales" aggregator="sum"
        formatString="$#,###.00"/>
```

Auflistung 11: Kennzahlen "Store Sales"

Änderungen index.html

In der index.html Datei wurden prinzipiell alle Funktionen die nicht die Auswahl von Ebenen und Kennzahlen und das einfache Setzen von Filtern in diesen Ebenen, durch Klick auf die Ebene selbst betrifft entfernt. Da nur so sichergestellt ist, dass die Probanden in der Benutzerstudie die gleiche Vorgehensweise bei der Lösung der gegebenen Aufgaben verwenden. Außerdem wurden die Funktionen „Abfrage Starten“ und „Achse Vertauschen“ belassen da es sich dabei um essentielle handelt.

Um den angesprochenen Namenskonventionen zu entsprechen wurden die beiden Templates „template-dimensions“ und „template-measures“ so verändert, dass sie den Namen einer Hierarchie bzw. Kennzahl nur bis vor einem Unterstrich anzeigen. Auflistung 12 zeigt einen Teil des Templates „template-measures“, dabei ist der geänderte Teil grün gekennzeichnet. Die Änderung im Template „template-dimensions“ funktioniert analog und wird an dieser Stelle nicht gezeigt.

```
<% _.each(measures, function(measure) { %>
    <% var pos_m = measure.caption.search("_"); %>
    <% if (pos_m >= 0) { %>
            <li><a title="<%= measure.description ? measure.description : measure.uniqueName
                %>"
                    rel="<%= measure.uniqueName %>"
                    class="measure" href="#Measures/member/<%= measure.uniqueName %>" >
                        <%=measure.caption.slice(0,pos_m) %></a>
            </li>
    <% } else { %>
            <li><a title="<%= measure.description ? measure.description : measure.uniqueName
                %>"
                    rel="<%= measure.uniqueName %>"
                    class="measure" href="#Measures/member/<%= measure.uniqueName %>">
                        <%= measure.caption %></a>
                        </li>
    <% } %>
<% }); %>
```

Auflistung 12: Verändertes Template „template-measures" in index.html

Ein Vorher- Nachher-Vergleich ist durch das Betrachten von Abbildung 6 und Abbildung 16 möglich.

Änderungen Settings.js

In der Settings.js Datei werden unter anderem die Query Eigenschaften gesetzt, hier wurde die Eigenschaft von automatischem Durchführen von Abfragen auf „false“ gesetzt, da ansonsten der automatische Austausch von Ebenen und Kennzahlen nicht funktionieren würde und diese Funktion generell nicht ideal bei größeren Datenmengen ist, da so bei mehreren Anwendern relativ schnell die Server Verbindung ausgelastet werden könnte.

```
QUERY_PROPERTIES: {
    'saiku.olap.query.automatic_execution': 'false',
    'saiku.olap.query.nonempty': 'true',
    'saiku.olap.query.nonempty.rows': 'true',
    'saiku.olap.query.nonempty.columns': 'true'
},
```

Auflistung 13: Saiku „QUERY_PROPERTIES" in Settings.js

Änderungen View DimensionList.js

DimensionList.js kontrolliert das Erscheinen und Verhalten der Dimensionen und Kennzahlen. Es reagiert dabei auf Click-Events am Hierarchie- und Kennzahlenbaum und ändert darauf hin das Erscheinungsbild und legt die ausgewählte Ebene oder Kennzahl, je nach Bedingung, auf das Spalten oder Zeilenfeld in der Arbeitsfläche von Saiku. Da diese Funktionalität genutzt wird um Ebenen und Kennzahlen auszutauschen, wenn der Anwender durch seine Auswahl in einen Würfelbereich gelangt in welchem dies nötig ist, wurde hier für den Fall, dass ein Click-Event nicht vom Anwender sondern von „HeteroHomogenousRuleset“ ausgelöst wird dieses Verhalten dahingehend verändert, dass in der Funktion „select_dimension“ bzw. „select_measure“ die Ebene oder Kennzahl dahin gelegt wird, wo sich die vorher gültige befunden hat. Auflistung 14 zeigt dies für die Funktion „select_measure“.

```
    if(!this.workspace.query.hhr.update){
        if(this.workspace.query.hhr.target == "COLUMNS"){
            $axis = $(this.workspace.el).find(".columns ul");
        }
        else{
            $axis = $(this.workspace.el).find(".rows ul");
        }
    }
    else{
        if ($(this.workspace.el).find(".rows ul .d_measure").length > 0) $axis =
            $(this.workspace.el).find(".rows ul");
        else if ($(this.workspace.el).find(".columns ul .d_measure").length > 0) $axis =
            $(this.workspace.el).find(".columns ul");
        else if ($(this.workspace.el).find(".filter ul .d_measure").length > 0) $axis =
            $(this.workspace.el).find(".filter ul");
        else $axis = $(this.workspace.el).find(".columns ul");
    }
```

Auflistung 14: Teile der Funktion "select_measure" in "DimensionList.js"

Änderungen Model Query.js

Query.js wird initialisiert sobald ein Würfel ausgewählt wurde, daher wird auch hier direkt, wie in Auflistung 15 zu sehen, HeteroHomogenousRuleset mit dem Workspace initialisiert um notwendige Informationen wie den ausgewählten Würfel auslesen zu können und später selbst automatisch Abfragen zu starten, da Query an den Workspace gebunden ist.

Die Funktion „parse“ in Query wird ebenfalls aufgerufen nachdem ein Anwender einen Würfel ausgewählt hat, daher wird an dieser Stelle die Sichtbarkeit der Hierarchien und Kennzahlen für das minimal homogene Schema gesetzt, das bei der Initialisierung von HeteroHomgenousRuleset von einem Webservice abgefragt wurde.

In der Funktion „run“ von Query wird immer die „update_hh“ Funktion aufgerufen sofern diese bereit ist für ein Update. Dies funktioniert da automatische Abfragen deaktiviert wurden. Wäre dies nicht der Fall könnte die Anwendung in einer Endlosschleife enden, da dann durch „update_hh“ unter Umständen auch eine Abfrage gestartet wird wenn eine Ebene oder Kennzahl ausgetauscht wird.

```
var Query = Backbone.Model.extend({
    initialize: function(args, options) {
        this.hhr = new HeteroHomogenousRuleset(options.workspace);
        …
    },
    parse: function(response) {
        …
        if(this.hhr.update){
            this.hhr.set_visibility(0, 0);
        }
        …
    },
    …
    run: function(force, mdx) {
        …
        if(this.hhr.update){
            this.hhr.update_hh();
        }
        if ((this.properties.properties['saiku.olap.query.automatic_execution'] == "false")
           &&
            !(force === true)) {
            return;
        }
        …
    },
    …
});
```

Auflistung 15: Funktionen in Query.js

Änderungen Model QueryAction.js

QueryAction verarbeitet genau drei Arten von Aktionen und zwar die Auswahl von Filtern, das Tauschen der Achsen und vom Anwender geschriebene MDX Abfragen. Da die Möglichkeit selbst MDX Abfragen zu schreiben für den Anwender nach den durchgeführten Änderungen an Saiku nicht mehr möglich ist, bleiben nur die beiden erstgenannten Aktionen übrig. Von diesen beiden Aktionen sind nur die gewählten Filter interessant, da anhand dieser entschieden werden muss, welche Hierarchien und Kennzahlen für den Anwender verfügbar sein müssen. Aus diesem Grund wird wie in Auflistung 16 zu sehen ist „action“ und „options“ an die Funktion „set_filter_selections“ in HeteroHomogenousRuleset übergeben. „action“ ist ein String der angibt ob die Ebene auf die ein Filter gesetzt wurde auf einer Zeile oder einer Spalte liegt und um welche Dimension es sich handelt.

Dieser String sieht beispielsweise so aus:

```
/axis/COLUMNS/dimension/Product
```

„options“ ist ein Objekt, das ein Array mit den ausgewählten Elementen der Ebene enthält die vom Anwender ausgewählt wurden.

```
put: function(action, options) {
    if(action != "/swapaxes" && this.query.hhr.update){
        this.query.hhr.newHierarchieNecessary = true;
        this.query.hhr.set_filter_selections(action, options);
    }
    …
},
```

Auflistung 16: Funktion „put" in QueryAction.js

HeteroHomogenousRuleset.js

HeteroHomogenousRuleset ist das wichtigste Objekt für die Umsetzung des hetero-homogenen Ansatzes in Saiku 2.4 und wurde als Backbone Model deklariert, dies ist nicht zwingend notwendig vereinfacht aber den Aufbau. Das Objekt besteht aus diesen Funktionen:

- initialize
- set_visibility
- set_filter_selections
- check_filter_selection
- update_hh
- replace_selections
- replace_filter_selections

Wie bereits erwähnt, wird HeteroHomogenousRuleset initialisiert sobald Query initialisiert wird und bindet sich dabei an Workspace um abfragen zu können welchen Würfel ein Anwender gewählt hat und später Aktionen zum Austauschen von Elementen, Kennzahlen und Filterselektionen setzen zu können. Auflistung 17 zeigt wie HeteroHomogenousRuleset initialisiert wird. Nach der Bindung an Workspace wird ein leeres Array für die Filterauswahl des Anwenders erzeugt, eine boolesche Variable „update“, die angibt ob ein Update der Auswahl und Filter durchgenommen werden darf, da unterscheidbar sein muss ob Änderungen durch den Anwender oder das System durchgeführt wurden. Es folgen die weiteren Variablen „dimensionList“ und „newHierarchieNecessary“, die beim Austausch von Filtern und Ebenen zu Kontrollzwecken benötigt werden. Die nächste Variable „cube“ speichert den vom Anwender verwendeten Datenwürfel und wird benötigt für Abfragen an den Webservice, der zu diesem Würfel je nach Auswahl von Werten in den Ebenen nach gültigen Hierarchien und Kennzahlen sucht und diese zurückgibt. Nachdem dieser Würfel ermittelt wurde, wird eine Ajax Abfrage an den Webservice ohne Ebenen und ausgewählte Werte gesendet, um das minimal homogene Schema des Würfels zu erhalten und in „minimalScheme“ zu speichern. Dies wird aus Gründen der Effizienz gemacht, da so keine Abfragen an den Webservice gesendet werden müssen wenn der Anwender keine Filter gesetzt hat aber eine Abfrage an den Saiku Server absendet.

Das Datenobjekt welches an den Webservice gesendet wird ist ein JSON Objekt und hat dabei diesen exemplarischen Aufbau:

[
{cube: “Sales_1997”, selection_filter:
[
{hierarchy_level: “[.]”,selection_values: “[]”}
]
}
]

Wie dieses JSON Objekt aussieht, wenn Werte in Ebenen zum Filtern des Ergebnisses ausgewählt wurden, wird später noch gezeigt.

Die Variable „hierarchiesAndMeasures“ wird bei der Initialisierung mit dem minimalen Schema gleichgesetzt. „previousHierarchiesAndMeasures“ wird mit „null“ initialisiert und später dazu verwendet um zu prüfen welche Hierarchien und Kennzahlen nach einer Abfrage an den Webservice gleich geblieben sind und diese nicht zu entfernen und neu hinzuzufügen.

```
initialize: function(workspace){
    this.ws = workspace;
    this.filter_selections = new Array();
    this.update = true;
    this.dimensionList = null;
    this.newHierarchieNecessary = true;
    this.cube = this.ws.selected_cube.substring(this.ws.selected_cube.lastIndexOf('/')+1,
                this.ws.selected_cube.length);
    this.minimalScheme = $.parseJSON(jQuery.ajax({
                                            url: 'http://xxx/HHDWService/
                                            getApplicableSchema/HierarchiesAndMeasures',
                                            type: 'POST',
                                            contentType:  "application/json",
                                            data: '[{"cube":' + '"' + this.cube + '"'+
                                            ',"selection_filter":[{"hierarchy_level":"[.]",
                                            "selection_values":[]}]}]',
                                            async: false,
                                            dataType: 'json'
                                    }).responseText);
    this.hierarchiesAndMeasures = this.minimalScheme;
    this.previousHierarchiesAndMeasures = null;
}
```

Auflistung 17: Initialisierung von HeteroHomogenousRuleset

Nach der Initialisierung des Query Objekts wird darin auch die Funktion „parse“ aufgerufen, welche wiederum die Funktion „set_visibility“ von HeteroHomogenousRuleset mit den Parametern (0, 0), wie in Auflistung 15 zu sehen, aufruft und diese dadurch dazu veranlasst die Sichtbarkeit der Hierarchien und Kennzahlen dem minimal homogenen Schema entsprechend zu setzen.

```
var selected_measures = new Array();
$("div[title=COLUMNS|ROWS] > .fields_list_body > .connectable > .ui-draggable >
.measure").each(function(idx,val){
       selected_measures[idx] = new Array();
       selected_measures[idx].push($(val).attr("rel"));
       seleced_measures[idx].push($(val).parent().parent().parent().parent().attr("title"));
});

var selected_levels = new Array();
$("div[title=COLUMNS|ROWS] > .fields_list_body > .connectable > .ui-draggable >
.level").each(function(idx,val){
       var h = $(val).attr("rel");
       selected_levels[idx] = new Array();
       selected_levels[idx].push(h.slice(h.indexOf('['),h.indexOf(']')+1));
       selected_levels[idx].push(h.slice(h.lastIndexOf('['),h.lastIndexOf(']')+1));
       selected_levels[idx].push($(val).parent().parent().parent().parent().attr("title"));
});
```

Auflistung 18: Aufnehmen von Ebenen und Kennzahlen in "update_hh"

Sobald der Anwender die ersten Ebenen und Kennzahlen auswählt wird „update_hh“ aufgerufen, solange dieser aber keine Filter setzt passieren keine Veränderungen aus der Sicht des Anwenders. Es wird allerdings, beim ersten Aufruf die Variable „dimensionList“ mit allen Dimensionen und Hierarchien gefüllt. Außerdem werden jedes Mal die gesetzten Ebenen und Kennzahlen in zwei Arrays aufgenommen um diese, im Fall das ein Filter gesetzt wird, und durch diesen der Anwender in eine neue Würfelregion gelangt austauschen zu können. Auflistung 18 zeigt diesen Vorgang.

Setzt ein Anwender einen Filter wird dieser über die Funktion „put“ in QueryAction an die Funktion „set_filter_selections“ in HeteroHomogenousRuleset mit den in Auflistung 16 beschriebenen Parametern gesendet. Falls dieser Filter noch nicht vorhanden ist wird er in das „filter_selections“ Array das bei der Initialisierung des HeteroHomogenousRuleset Objekts erstellt wurde angehängt.

Dieses Array sieht so aus wenn in der Dimension „Store“ die Ebene „Country“ auf USA gefiltert wurde:

[
{hierarchy: “[Store.Store_ALL]”, uniquename: “[Store.Store_ALL].[Store Country]”, type: “level”},
{uniquename: “[Store.Store_ALL].[USA]”, type: “member”},
{axis: “/axis/COLUMNS/dimension/Store”}
]

Das erste JSON Objekt in dem Array gibt an, dass es sich um die Hierarchie „Store_ALL“ in der Dimension „Store“ handelt und die gewählte Ebene in dieser Hierarchie „Country“ ist. Das zweite JSON Objekt in dem Array gibt an, dass das gewählte Element aus der Ebene „Country“ in der Hierarchie „Store_ALL“ aus der Dimension „Store“ den Wert „USA“ hat. Diese Strukturierung wird von Saiku 2.4 vorgegeben und später bei der weiteren Versendung an den Webservice, während der Ausführung der Funktion „update_hh“, vereinfacht. Bevor dies passiert wird aber mit der Funktion „check_filter_selection“ geprüft, ob die jeweiligen Filter noch korrekt sind und gegebenenfalls gelöscht um keine falschen zu setzen oder Filter auf Ebenen zu setzen, die gar nicht mehr ausgewählt sind. Danach folgt der vereinfachte Aufbau des JSON Objekts das an den Webservice gesendet wird um auf Grund der Filter die richtigen Hierarchien und Kennzahlen zu erhalten und anschließend in der Benutzersicht die richtigen Filter einzusetzen. Auflistung 19 zeigt, wie dies in der „update_hh“ Funktion gelöst wird. Es werden aus der Variable „filter_selections“ die Hierarchien und ausgewählte Ebenen und die Werte zur Filterung in dieser Ebene extrahiert und dem Array „selections“ hinzugefügt. Dieses Array wird danach zusammen mit dem Würfel als JSON Objekt in der Variable „hhr_selections“ gespeichert welche wieder in einem Array gespeichert wird um per AJAX Aufruf an den Webservice gesendet zu werden. Dieser gibt die zu dieser Auswahl gültigen Hierarchien und Kennzahlen zurück die daraufhin in der Variable „hierarchiesAndMeasures“ gespeichert werden.

```
var selections = new Array();
for(var i = 0;i<this.filter_selections.length;i++){
     var values = new Array();
     for(var j = 1;j<this.filter_selections[i].length;j++){
          var val = this.filter_selections[i][j].uniquename;
          if(this.filter_selections[i][j].type === "member"){
                values.push(val.slice(val.lastIndexOf('['),val.length));
          }
     };
     selections.push({
           hierarchy_level: this.filter_selections[i][0].uniquename,
           selection_values: values
     });
}

var hhr_selections = {
     "cube": this.cube,
     "selection_filter": selections
}

var hhr_selections_array = new Array(hhr_selections);
…
this.hierarchiesAndMeasures = $.parseJSON(jQuery.ajax({
                                   url:
'http://xxx/HHDWService/getApplicableSchema/HierarchiesAndMeasures',
                                   type: 'POST',
                                   contentType:  "application/json",
                                   data: JSON.stringify(hhr_selections_array),
                                   async: false,
                                   dataType: 'json'
                                 }).responseText);
this.replace_selections(selected_levels, selected_measures);
```

Auflistung 19: Erstellung der Abfrage für Webservice

Die Variable „hhr_selections“ hat, ausgehend von dem zuvor gegebenen Beispiel mit der Dimension „Store“, der Ebene „Country“ und der Auswahl „USA“, diesen Aufbau:

[
{cube: “Sales_1997”,
selection_filter:
[
{hierarchy_level: “[Store.Store_ALL].[Store Country]”,
selection_values: “[USA]”}
]
}
]

Das Ergebnis dieser Abfrage an den Webservice gibt neben den minimal homogenen Hierarchien und den minimal homogenen Kennzahlen eine neue Hierarchie anstatt der Hierachie „[Store.Store_ALL]“ zurück, nämlich die Hierarchie „[Store.Store_USA]“. Dieses Ergebnis sieht so aus:

{hierarchies: [hierarchy: “[Store_USA]“, hierarchy: “[Customers.Customers_ALL]“, …],
measures: [measure: “[Store Sales_ALL]”, measure: “[Store Profit_ALL]”, ...]}

Da bei dem Beispiel eine Ebene ausgetauscht wird und der Filter neu gesetzt wird, wird die Funktion „replace_selections“ aufgerufen. Diese prüft zunächst alle ausgewählten Ebenen und Kennzahlen dahingehend, ob die jeweilige Hierarchie noch dieselbe ist wie in „hierarchiesAndMeasures“, sollte dies nicht der Fall sein tauscht sie diese aus indem sie die Ebene bzw. die Kennzahl löscht und mit der neuen Hierarchie wieder einfügt. Auflistung 20 zeigt wie eine Ebene eingefügt wird.

```
for(var i = 0; i < l.length; i++){
    this.target = l[i][2];
    if(typeof ($("a[title=\"" + l[i][0] + "." + l[i][1] + "\"]")[0]) !== "undefined"){
         this.ws.dimension_list.select_dimension({
            target: $("a[title=\"" + l[i][0] + "." + l[i][1] + "\"]")
         });
    }
}
```

Auflistung 20: Austausch von Ebenen

Zum Einfügen der Ebene wird die Funktion „select_dimension“ verwendet, die in dem View „dimension_list“ enthalten ist, welcher an das Model „ws“ (Workspace) gebunden ist und eingesetzt werden kann, da bei der Initialisierung von HeteroHomogenousRuleset das Workspace Model mit diesem verbunden wurde.

Da durch das Entfernen der Ebenen und Neusetzen die Filter in diesen Ebenen nicht mehr gelten, müssen diese neu gesetzt werden, was durch den Aufruf der Funktion „replace_filter_selections“ geschieht. Diese Funktion prüft zuerst die einzelnen Ebenen ob die alten Hierarchien „previousHierarchiesAndMeasures“ ident mit den neuen sind oder nicht und tauscht anschließend im Array „filter_selections“ diese gegebenenfalls aus und verwendet dann für diese Ebenen die „put“ Funktion des QueryAction Models wie in Auflistung 21 zu sehen.

```
if(validFilter){this.ws.query.action.put(this.filter_selections[i][this.filter_selections[i].l
ength-1].axis, {
    success: function(){},
        data: {
            selections: JSON.stringify(replaceFilterSelection)
        }
    });
}
```

Auflistung 21: Filter neu setzen

Nach diesen Aktionen ist der Update Prozess abgeschlossen und die „set_visibility“ Funktion wird wieder aufgerufen um die jetzt gültigen Hierarchien und Kennzahlen anzuzeigen und die nicht mehr gültigen auszublenden.

5. Studie

Dieses Kapitel beschreibt die Studie, die für diese Arbeit durchgeführt wurde um die Usability des hetero-homogenen Ansatzes in OLAP-Anwendungen zu testen und damit die Nutzbarkeit dieses Ansatzes zu überprüfen. Außerdem wird zu Kontrollzwecken dieselbe OLAP-Anwendung mit einem rein heterogenen Ansatz geprüft. Kapitel 5.1 beschreibt das Untersuchungsdesign, Kapitel 5.2 zeigt die Ergebnisse der Studie und in Kapitel 5.3 folgt die Interpretation.

5.1. Untersuchungsdesign

In diesem Unterkapitel werden die Methoden und die Anpassungen die an diesen vorgenommen wurden beschrieben. Die Testumgebung wird vorgestellt und alle wichtigen Informationen zur Durchführung selbst werden gegeben.

5.1.1. Methoden

Wie bereits in Kapitel 2.3.2 beschrieben wurde für diese Studie der SUS-Fragebogen [15] verwendet, den die Probanden nach Abschluss aller Aufgaben beantworten mussten. Außerdem mussten sie nach der Lösung einer Aufgabenstellung die Schwierigkeit dieser auf einer Likert-Skala von 1-7 [14] bewerten um sehen zu können, in welchen Bereichen die Probanden eventuell Probleme haben. Neben diesen beiden Methoden, die einen subjektiven Charakter haben, wurden die Arbeitsschritte der Probanden durch Piwik [24], eine Webanalytic-Anwendung auf Open-Source-Basis die ähnlich wie Google Analytics arbeitet, aufgezeichnet, um die Ergebnisse besser interpretieren zu können. Um einerseits den Fragebogen schneller auswerten zu können und andererseits die Bedeutung der individuellen SUS-Scores besser zu interpretieren wurde dem SUS-Fragebogen außerdem noch eine 11. Frage angehängt die die Anwendung mittels subjektiver Bewertung auf einer siebenstufigen Likert-Skala bewerten soll und laut Studien [25] hochgradig mit der SUS-Score korreliert bei einem $\alpha < 0.01$ und $r = 0.822$.

Da die Fragebögen und Fragen im Original auf Englisch sind, wurden diese soweit vorhanden, von anderen Studien die diese Methoden in deutscher Sprache verwendet haben übernommen. Die Fragen für den SUS-Fragebogen lauten in deutscher Sprache wie folgt [26]:

1. Ich denke, dass ich dieses System gerne regelmäßig nutzen würde.
2. Ich fand das System unnötig komplex.
3. Ich denke, das System war leicht zu benutzen.
4. Ich denke, ich würde die Unterstützung einer fachkundigen Person benötigen, um das System benutzen zu können.
5. Ich fand, die verschiedenen Funktionen des Systems waren gut integriert.
6. Ich halte das System für zu inkonsistent.
7. Ich glaube, dass die meisten Menschen sehr schnell lernen würden, mit dem System umzugehen.
8. Ich fand das System sehr umständlich zu benutzen.
9. Ich fühlte mich bei der Nutzung des Systems sehr sicher.
10. Ich musste viele Dinge lernen, bevor ich mit dem System arbeiten konnte.

Für die siebenstufige Likert-Skala zur Bewertung der Anwendung konnte keine verwendete deutsche Übersetzung gefunden werden, sie wurde daher selbst übersetzt. Das Original lautete [25]:

Overall, I would rate the user-friendliness of this product as:

Worst Imagi-nable	Awful	Poor	OK	Good	Excellent	Best Imagi-nable

Eigene Übersetzung:

Im Gesamten würde ich die Nutzerfreundlichkeit dieser Anwendung folgendermaßen bewerten:

Schlechtest möglich	Sehr schlecht	Schlecht	OK	Gut	Sehr Gut	Best möglich

Die Frage zur Bewertung der Schwierigkeit einer einzelnen Aufgabe [14] in englischer Sprache wurde ebenfalls selbst übersetzt und adaptiert und lautete im Original:

„Overall, this task was:"

	1	2	3	4	5	6	7	
Very difficult								Very easy

Eigene Variante:

Wie schwierig haben Sie diesen Task gefunden (1 = sehr schwierig, 7 = sehr leicht)?

1	2	3	4	5	6	7

Wie die SUS-Score ermittelt wird wurde bereits in Kapitel 2.3.2. erläutert, Abbildung 13 zeigt die Bedeutung der SUS-Score und der mit diesem Ergebnis stark korrelierenden siebenstufigen Likert-Skala zur Usability einer Anwendung („Schlechtest möglich" und „Sehr schlecht" sind zusammengefasst):

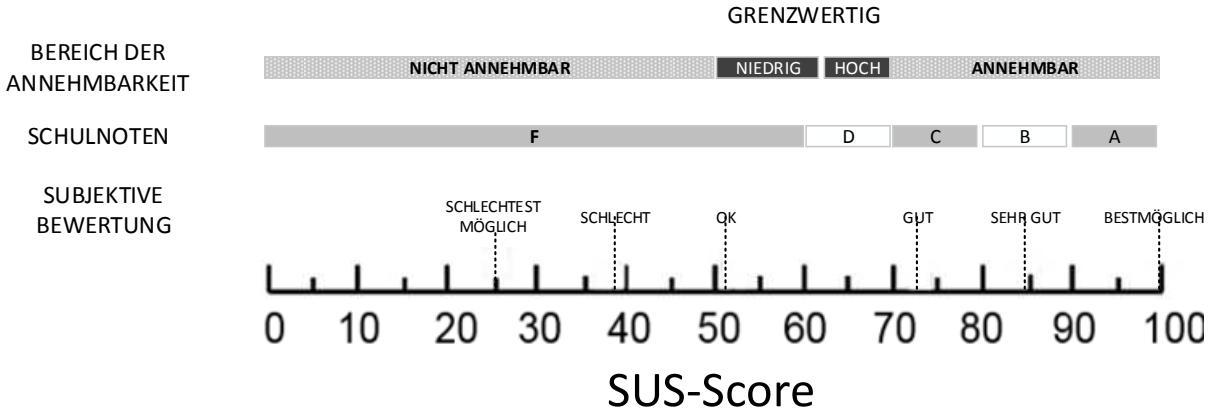


Abbildung 13: Vergleich der subjektiven Bewertung zu SUS-Score [25]

5.1.2. Testumgebung

In diesem Kapitel wird die Testumgebung der Studie beschrieben, darunter ist hier der technische Aufbau zu verstehen. Zuerst werden kurz die Hardware-Spezifikationen angeführt und anschließend die verwendeten Softwarekomponenten und deren Aufbau. Danach wird das für die Studie verwendete multidimensionale Datenmodell gezeigt.

Hardware

Wie bereits zu Beginn dieser Arbeit beschrieben, handelt es sich bei dem verwendeten OLAP-Tool um eine webbasierte Lösung. Für den serverseitigen Teil des hetero-homogenen wie auch des heterogenen OLAP-Tools, den Webserver zum Tracken der Benutzer und die Datenbank mit den Dimensions- und Faktentabellen wurde ein Rechner mit einem Pentium E6700 Dual-Core Prozessor mit 3,2 GHz Taktfrequenz und 2 MB Cache verwendet, der Arbeitsspeicher lag bei 4 GB.

Da für das hetero-homogene OLAP-Tool wie in Kapitel 4 beschrieben auch noch eine Oracle Datenbank benötigt wird, ist diese auf einem virtuellen Server aufgesetzt, der mit der Leistung eines Xeon x5650 mit 6 Kernen auf 2,66 GHz Taktfrequenz und 12 MB Cache und 8 GB Arbeitsspeicher arbeitet.

Software

Als Betriebssystem wird bei beiden vorhin angesprochenen Rechnern CentOS 6.6 verwendet, wobei auf dem virtuellen Server auch die Xfce Desktop Umgebung in der Version 4.8.0 läuft um die Oracle Datenbank in Version 11g R2 installieren zu können. Genaue Installations- und Konfigurationsanweisungen werden an dieser Stelle nicht gegeben, es wird für die jeweiligen Anwendung die nicht speziell verändert wurden auf die Herstellerseite Verwiesen und für eigene Anwendungen auf den Punkt Installation im Anhang.

Diese Softwarekomponenten sind für den Rechner auf dem die serverseitigen Komponenten des OLAP-Tools installiert sind notwendig:

- MySQL 5.1.73
- Apache Tomcat 6.0.24 und Saiku 2.4 heterogen
- Apache Tomcat 6.0.24 und Saiku 2.4 hetero-homogen mit HHDWService
- MySQL Connector/J 5.1.*
- Oracle Database 11g R2 JDBC Treiber ojdbc6
- Java 1.7.0_72 64 bit
- Apache 2.2.15
- Piwik 2.9.1

Diese Komponenten wurden auf dem zweiten Server installiert und sind nur für das hetero-homogene OLAP-Tool relevant:

- Oracle 11g R2
- Hetero-Homogeneous Data Warehouse [3]

In Abbildung 14 und Abbildung 15 wird schematisch die grundsätzliche Funktionalität der Testumgebung dargestellt, die Abbildung 14 zeigt diese für das heterogene-, Abbildung 15 für das hetero-homogene Testsetup.

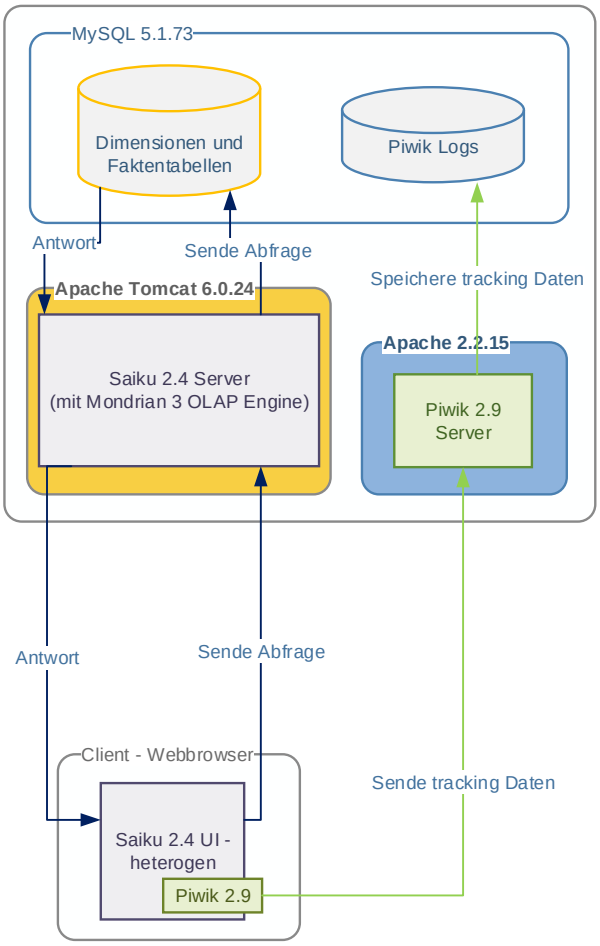


Abbildung 14: Testumgebung heterogenes OLAP

Abbildung 14 zeigt schematisch den Aufbau des Systems. Sendet der Anwender über seinen Browser, in dem das OLAP-Frontend in der heterogenen Variante läuft, eine Abfrage an den Server, so durchläuft diese Abfrage mehrere Operationen auf dem OLAP-Backend, das selbst die Daten für die vom Anwender gestellte Anfrage von der MySQL Datenbank in der die Dimensionen und Faktentabellen des Data Warehouse gespeichert sind abfragt und anschließend das Ergebnis an den Anwender zurücksendet. Während der Anwender Manipulationen vornimmt (also Kennzahlen und Ebenen für die Abfrage auswählen, diese tauschen, anordnen, löschen usw.) werden diese Aktionen von Piwik, einer Webanalyse-Anwendung, aufgezeichnet und an den Piwik Server gesendet der diese wiederum in einer MySQL Datenbank abspeichert. Abbildung 15 zeigt den gleichen Prozess mit dem hetero-homogenen Ansatz, prinzipiell funktioniert dieser gleich, nur wird immer wenn der Anwender einen Filter auf ausgewählten Ebenen setzt dieser an den HHDWService gesendet der prüft ob bei dieser Auswahl möglicherweise neue Ebenen und Kennzahlen verfügbar sind.

Dieser Webservice bezieht die Informationen aus der Oracle Datenbank. Sollten neue Ebenen und Kennzahlen verfügbar sein, werden diese für den Anwender sichtbar gemacht.

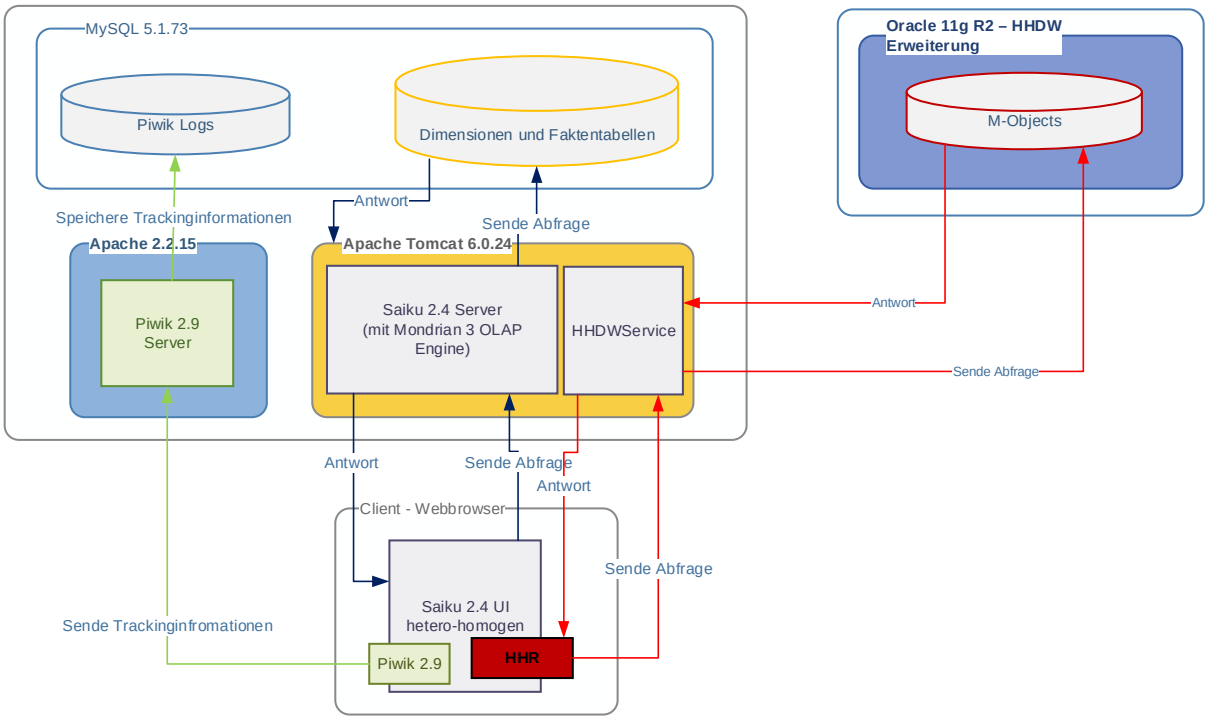


Abbildung 15: Testumgebung hetero-homogenes OLAP

Von beiden Varianten des OLAP-Tools wurden alle Funktionalitäten, die nicht notwendig sind um durch einen Datenwürfel zu navigieren und Abfragen durchzuführen, aus dem Frontend entfernt um sicherzustellen, dass die Usability des Suchvorgangs selbst bewertet wird und nicht bestimmte Fähigkeiten die das System ansonsten bietet. Des Weiteren wurde ein Button „Task abgeschlossen“ hinzugefügt, den die Probanden nach jedem Task drücken müssen um später feststellen zu können welche Aktionen für welchen Task getätigt wurden. Abbildung 16 zeigt das OLAP-Tool nach den durchgeführten Änderungen.

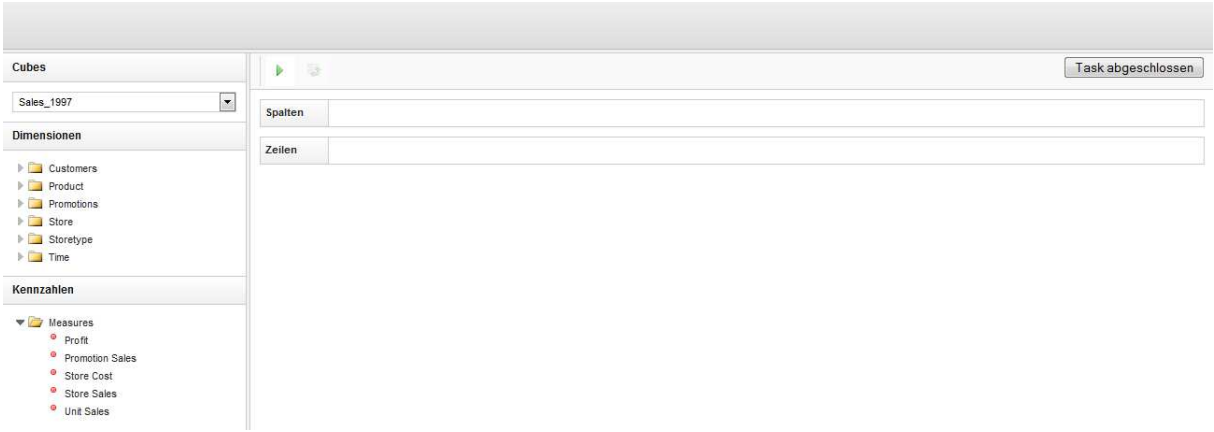


Abbildung 16: Saiku 2.4 nach Änderungen

Bei der Anwendung um die Aktionen der Probanden zu aufzuzeichnen, handelt es sich um eine Webanalyse-Plattform die sich Piwik nennt und ähnliche Fähigkeiten wie Google Analytics besitzt. Piwik wurde ausgewählt, da es sich um Open-Source-Software handelt die sehr einfach zu verwenden ist und die aufgezeichneten Daten in einer eigenen Datenbank gespeichert werden können und damit die Datensicherheit gewahrt werden kann. Außerdem ist diese Plattform einfach konfigurierbar und liefert Reports bereits sehr gut in der Form aufbereitet, die eine schnelle Analyse der Handlungen der Probanden erlaubt [24]. Abbildung 18 zeigt einen Teil eines Benutzerlogs der in dieser Form direkt über einen Webbrowser abrufbar ist.

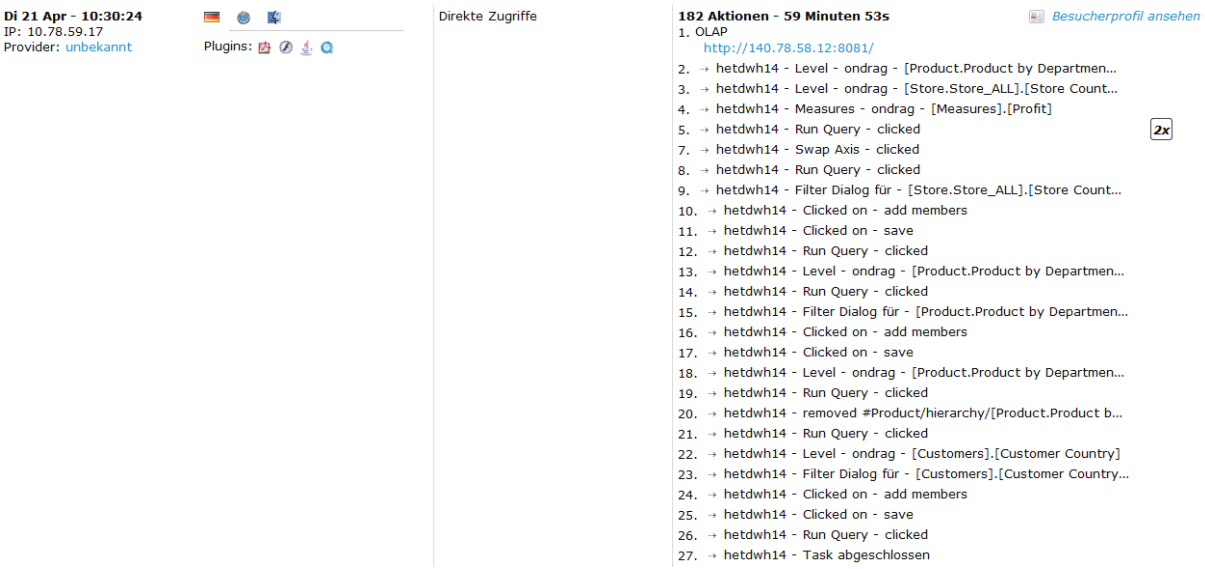


Abbildung 17: Piwik Benutzerlog

Um Piwik verwenden zu können müssen nach der Installation auf einem Apache Webserver noch für die Seiten, die getrackt werden sollen, ein Trackingcode erzeugt und in diese Seiten kopiert werden. Außerdem müssen noch jene Elemente der Seite bzw. Anwendung markiert werden bei welcher Benutzeraktion welche Information an den Piwik Server gesendet werden soll. Diese Trackingcodes sind im Normalfall Javascript-Tags.

```
<!-- Piwik -->
        <script type="text/javascript">
          var _paq = _paq || [];
          _paq.push(["setDocumentTitle",document.domain + "/" + document.title]);
          _paq.push(['trackPageView']);
          _paq.push(['enableLinkTracking']);
          (function() {
            var u=(("https:" == document.location.protocol) ? "https" : "http") +
            "://xxx/piwik/";
            _paq.push(['setTrackerUrl', u+'piwik.php']);
            _paq.push(['setSiteId', 1]);
            var d=document, g=d.createElement('script'),
            s=d.getElementsByTagName('script')[0]; g.type='text/javascript';
            g.defer=true; g.async=true; g.src=u+'piwik.js'; s.parentNode.insertBefore(g,s);
          })();
        </script>
        <noscript><p><img src="http://xxx/piwik/piwik.php?idsite=1" style="border:0;" alt=""
/></p></noscript>
<!-- End Piwik Code -->
```

Auflistung 22: Piwik Trackingcode

Auflistung 22 zeigt den Trackingcode, dieser muss auf jene Seiten die getrackt werden sollen ans Ende des „body“ teils im HTML Dokument kopiert werden. Im Fall der Benutzerstudie ist dies jeweils die index.html des heterogenen sowie des hetero-homogenen OLAP-Tools. Auflistung 23 zeigt wie die Aktion aufgezeichnet wird wenn ein bestimmter Anwender eine Kennzahl aus dem Navigationsbaum des OLAP-Tools zieht oder ihn anklickt. Dabei wird bei „ondragstart“ oder „onclick“ ein Event ausgelöst, das Informationen in ein Array schreibt das danach an den Piwik Server gesendet wird der diese Informationen abspeichert. In diesem Fall sind die Informationen der Benutzername, das eine Kennzahl entweder angeklickt oder bei gedrückter Maustaste aufgehoben wurde um sie an einen anderen Platz zu ziehen - was ebenfalls getrackt wird sobald wieder niedergelegt - und die Bezeichnung der Kennzahl selbst.

```
<li ondragstart="javascript:_paq.push(['trackEvent', '<%= this.Saiku.session.username %> -
Measures - ondrag', '<%= measure.uniqueName %>']);"><a title="<%= measure.description ? meas-
ure.description : measure.uniqueName %>"
                                rel="<%= measure.uniqueName %>"
                              class="measure" href="#Measures/member/<%= measure.uniqueName %>"
onclick="javascript:_paq.push(['trackEvent', '<%= this.Saiku.session.username %> - Measures -
onclick', '<%= measure.uniqueName %>']);" ><%= measure.caption %></a>
</li>
```

Auflistung 23: Tracking einer Kennzahl bei Aktion "ondrag" oder "onclick"

Da Piwik aus Performance Gründen standardmäßig nur die Top 500 Seiten, Top 1000 Custom Variablen, Events, Aktionen usw. in den Reports speichert, und alle weiteren unter dem Label „andere“ sichert muss in diesem Fall, da dies relevant ist und davon auszugehen ist, dass insgesamt deutlich mehr Aktionen getätigt werden, geändert werden. Dazu wurden in der globalen Konfigurationsdatei von Piwik diese Werte auf 100.000 geändert um sicherzustellen, dass alle Aktionen ohne Ausnahmen sichtbar sind.

Datenmodell

Als Datenbasis für diese Benutzerstudie wurde das „Foodmart“ Beispiel welches von Pentaho [19] für Mondrian als Demobeispiel bereitgestellt wird verwendet und verändert um heterogene Eigenschaften einzuführen.

In dem Beispiel geht es um einen Supermarkt der Filialen in den USA, Mexiko und Kanada besitzt und dort übliche Güter verkauft, die im Sortiment eines klassischen Supermarkts zu finden sind. Da es länderspezifische Unterschiede, sowohl auf technologischer Ebene als auch politischer Ebene, gibt, diese Informationen aber behalten werden möchten, ist in einigen der Dimensionen und Fakten des Data Warehouse Heterogenität enthalten. Abbildung 19 zeigt das konzeptuelle Modell des heterogenen Würfels unter einer vereinfachten Verwendung des Dimensional Fact Models von Golfarelli et al. [27]. Aus Übersichtlichkeitsgründen werden Dimensionen die für die Aufgaben nicht benötigt werden nicht in der Abbildung gezeigt.

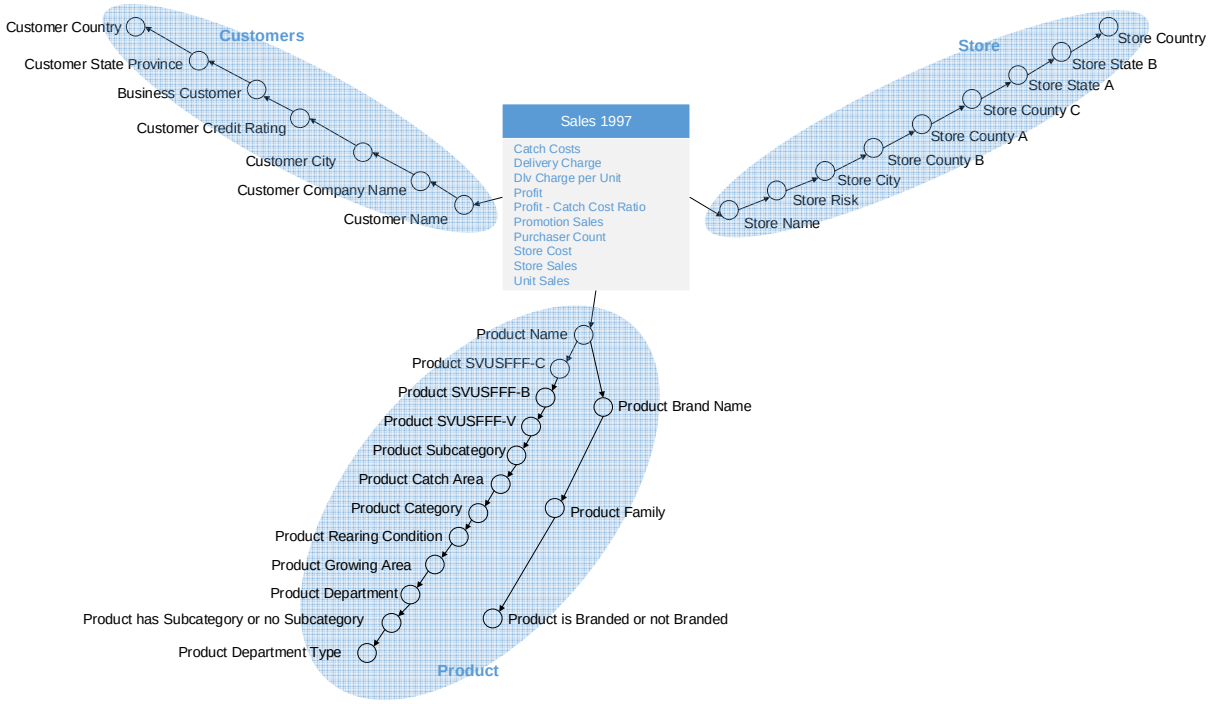


Abbildung 18: Heterogenes Modell des Würfels für die Studie

Abbildung 18 zeigt den Würfel „Sales 1997“ mit den Kennzahlen „Catch Cost“, „Delivery Charge“, „Dlv Charge per Unit“, „Profit“, „Profit – Catch Cost Ratio“, „Promotion Sales“, „Purchaser Count“, „Store Cost”, „Store Sales” und „Unit Sales”. Außerdem sind die drei Dimensionen „Customers“, „Store“ und „Product“ zu sehen. Nicht in der Abbildung zu sehen sind die Dimensionen „Store Type“, „Time“ und „Promotions“ da sie in den Aufgabenstellungen in der Studie nicht benötigt werden. Die Kennzahlen „Catch Cost“, „Delivery Charge“, „Dlv Charge per Unit“ und „Profit – Catch Cost Ratio“ sind hochgradig heterogen und in nur sehr kleinen Bereichen des Würfels vorhanden die in den Aufgabenstellungen der Studie ebenfalls nicht gesucht waren und werden daher hier näher erklärt.

Die Kennzahl „Purchaser Count“ ist nur in Geschäften der USA vorhanden, wenn dort Kunden mit einer Kundenkarte eingekauft haben, und daher eindeutig identifiziert werden können. In dem Beispiel ist die Bedingung, dass alle Kunden der USA eine Karte besitzen und immer mit dieser bezahlen. Die weiteren Kennzahlen sind in allen Bereichen vorhanden aber wieder bei Geschäften der USA mit Kunden aus den USA in feinerer Granularität bis auf den eindeutigen Kunden hinab.

In der „Customers“ Dimension gibt es darüber hinaus noch weitere Heterogenität, denn in Kalifornien gibt es einige Geschäftskunden und zu diesen sind Kreditinformationen und der Name des Geschäfts bekannt.

In der Dimension „Store“ sind die Ebenen „Store State A“ und „Store County A“ nur in den in den USA vorhanden, „Store State B“ und „Store County B“ nur in Kanada und „Store County C“ und „Store Risk“ nur in Mexiko.

In der Dimension „Product“ gibt es zwei alternative Hierarchien, eine die über den „Product Department Type“ aufrollt und eine zweite über „Product is Branded or not Branded“. Da manche Produkte keine Marke haben gibt es bei diesen auch keinen Markennamen, was eine Heterogenität in der „Product is Branded or not Branded“ Hierarchie bedeutet. In der Hierarchie mit „Product Department Type“ gibt es Heterogenität auf den Ebenen „Product has Subcategory or no Subcategory“, „Product Growing Area“, „Product Rearing Condition”, „Product Category“, „Product Subcategory” und den einzelnen Ebenen mit „SVUSFFF“ Ziffern. Das heterogene Mondrian-Schema zu dieser Datenlage ist im Anhang unter *Heterogenes Mondrian-Schema* zu finden. Das Mondrian-Schema für den hetero-homogenen Ansatz ist im Anhang unter *Hetero-homogenes Mondrian-Schema* zu finden. Abbildung

19, Abbildung 20 und Abbildung 21 zeigen wie die heterogenen Hierarchien als M-Objects dargestellt sind um dem hetero-homogenen Ansatz zu entsprechen und so in dieser Studie verwendet werden.

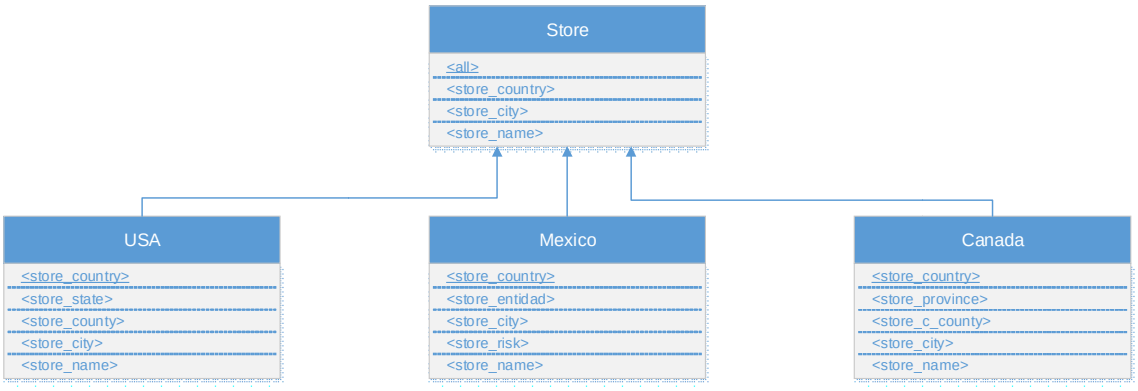


Abbildung 19: Hetero-homogene Hierarchien der Dimension „Store“

In Abbildung 19 ist das oberste M-Object die minimal homogene Hierarchie der Dimension „Store“ und beinhaltet die Ebenen des Landes, der Stadt und des Geschäftes. Diese Ebenen sind sowohl in den USA, Mexiko als auch Kanada gültig. Die weiteren M-Objetcs zeigen die Sub-Hierarchien die für sich wieder homogen sind aber im Gesamtkontext heterogen.

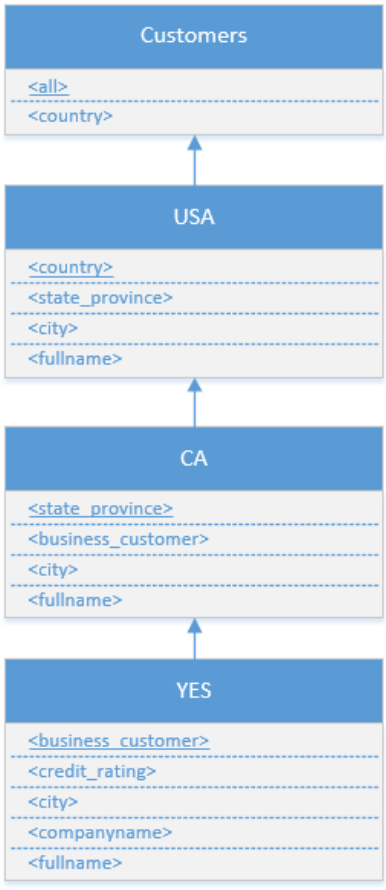


Abbildung 20: Hetero-homogene Hierarchie der Dimension „Customers"

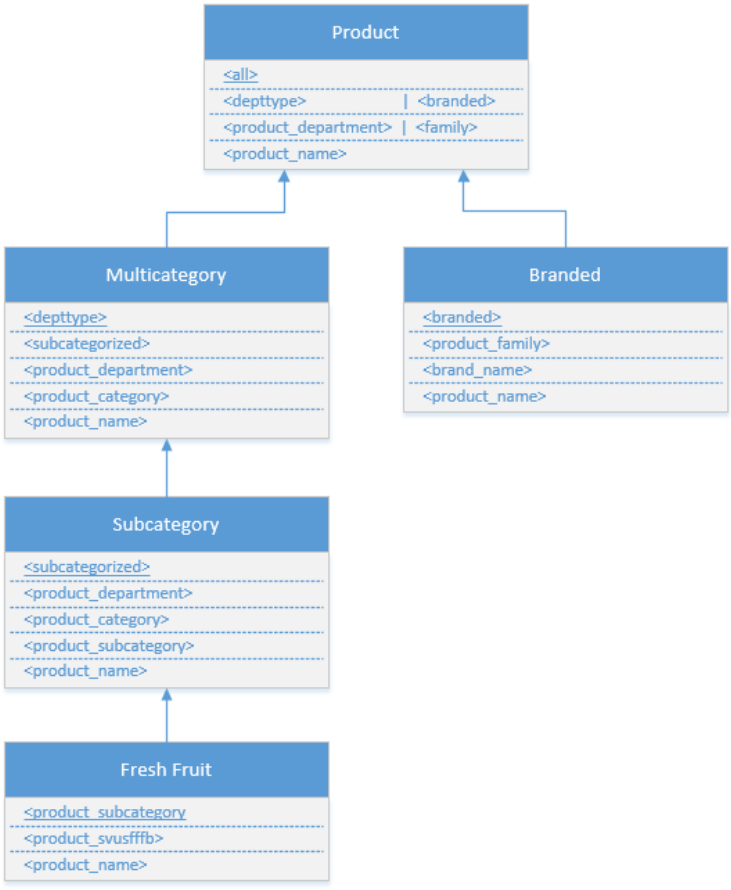


Abbildung 21: Hetero-homogene Hierarchien der Dimension "Product"

Abbildung 20 und Abbildung 21 zeigen die Hierarchien der M-Objects der Dimensionen „Customers“ und „Product“. Die Dimension „Product“ hat die alternativen Hierarchien, die in der Abbildung durch die Ausprägungen „Multicategory“ und „Branded“ erkennbar ist.

5.1.3. Durchführung

Da für valide Ergebnisse in Usability-Studien unter Einsatz des SUS-Fragebogens als Testmethode mindestens 12 Testpersonen empfohlen werden und auf Grund des Testszenarios, Personen mit betriebswirtschaftlichen oder volkswirtschaftlichen Kenntnissen nötig waren und es ebenfalls von Vorteil war, wenn diese Personen technische Kenntnisse im Datenbankbereich haben, wurden Wirtschaftsinformatik Studenten mit mittel- bis weit fortgeschrittenem Studium ausgewählt, welche die Lehrveranstaltung Data & Knowledge Engineering besuchen, da somit neben den angesprochenen notwendigen Kenntnissen auch der Zugang zu genug qualifizierten Probanden und benötigten Räumlichkeiten gegeben war. Die Lehrveranstaltung Data & Knowledge Engineering vermittelt erweiterte Kenntnisse von Datenbanksystemen.

Testgruppen

Durch die an der Universität gegebenen Bedingungen konnten drei Testgruppen gebildet werden. Da in dieser Studie in erster Linie die Usability des hetero-homogenen Ansatzes von Interesse ist wurde entschieden, dass zwei Gruppen mit dem OLAP-Tool arbeiten das diesen umsetzt und eine Gruppe mit dem OLAP-Tool das den heterogenen umsetzt, im Ergebnis werden die beiden Gruppen, welche mit dem hetero-homogenen Tool arbeiten mussten, zusammengefasst.

Die erste Gruppe bestand aus 21 Personen, die Studie wurde bei dieser um 8:30 am Morgen durchgeführt und betraf das hetero-homogene OLAP-Tool.

Die zweite Gruppe bestand ebenfalls aus 21 Personen, die Studie wurde bei dieser um 10:15 durchgeführt und betraf das heterogene OLAP-Tool.

Die dritte Gruppe bestand aus 24 Personen, die Studie wurde bei dieser um 13:45 am Nachmittag durchgeführt und betraf das hetero-homogene OLAP-Tool.

Die Studienteilnehmer mussten vor Beginn mit der Bearbeitung der Aufgaben und Bewertung der Anwendung noch demografische Angaben über Alter, Geschlecht, höchste abgeschlossene Ausbildung und momentanes Hauptstudium angeben.

Bei allen Gruppen wurde versucht jeden Probanden alleine an einem Tisch arbeiten zu lassen um zu verhindern, dass diese sich gegenseitig während der Lösung der Aufgaben beobachteten. Weitestgehend war dies möglich, allerdings mussten aus Platzgründen vereinzelt Personen in der dritten Gruppe nahe zusammen sitzen.

Einschulung OLAP-Tool

Da es nötig war den Probanden ein grundlegendes Basiswissen zum Umgang mit dem jeweiligen OLAP-Tool zu vermitteln wurden diese kurz im Umgang damit eingeschult. Dies funktionierte so, dass ein Studienleiter, anhand eines Drehbuchs, das im Anhang unter *Drehbuch – Anweisungen zu OLAP mit hetero-homogenen DW* bzw. *Drehbuch – Anweisungen zu OLAP mit heterogenen DW* zu finden ist, die notwendigen Funktionalitäten vortrug und ein zweiter Studienleiter dies auf einem Beamer vorzeigte, während die Probanden selbst mitarbeiten um sich an die Arbeit mit der Anwendung zu gewöhnen.

Aufgabenstellung

Um die Usability der jeweiligen Anwendung zu testen wurden für die in der Testumgebung beschriebene Datenbasis und das multidimensionale Schema als Aufgabenstellung 5 Tasks definiert die jeweils aus mehreren einzelnen Aufgaben bestehen und Suchprozesse abdecken in denen heterogene Hierarchieebenen und Kennzahlen vorkommen, die nur für bestimmte Datenbereiche Gültigkeit besitzen und sowohl einfache als auch komplexe Suchvorgänge erfordern. Die Formulierung und Ausgestaltung der Aufgabenstellungen wurde dabei durch mehrere Pretests mit Studenten und Bekannten verfeinert. Die gesamte Aufgabenstellung, inklusive Fragebogen ist unter *Aufgabenstellung und Fra-*

gebogen im Anhang zu finden. Die Tasks waren so aufgebaut, dass in den Tasks 1-3 noch sehr genau angegeben wurde wie die Probanden vorgehen müssen um an die gesuchte Information zu gelangen indem die notwendigen Ebenen und Filter vorgegeben wurden. Task 4 und 5 mussten frei ohne Zusatzinformationen gelöst werden mussten. Task 1 und Task 5 waren so aufgebaut, dass die Probanden in eine Würfelregion navigieren mussten die eine zusätzliche Kennzahl beinhaltet, in den anderen Aufgaben mussten nur eine oder mehrere Sub-Hierarchien mit zusätzlichen Ebenen betreten werden.

5.2. Ergebnisse

Dieses Kapitel vermittelt die Ergebnisse der Studie, eine Interpretation dieser folgt im nächsten Kapitel. Wie in Kapitel 5.1.3 angesprochen, gab es insgesamt 66 Teilnehmer in drei Gruppen. 21 Teilnehmer kamen dabei aus der ersten Gruppe und mussten mit dem hetero-homogenen OLAP-Tool arbeiten. Die zweite Gruppe, ebenfalls bestehend aus 21 Teilnehmern, arbeitete mit dem rein heterogenen OLAP-Tool und die dritte Gruppe, bestehend aus 24 Teilnehmern, arbeitete wieder mit dem hetero-homogenen OLAP-Tool. Die erste und die dritte Gruppe werden im Ergebnis zusammengefasst. Aus der ersten Gruppe mussten drei Teilnehmer aus der Bewertung ausgenommen werden, da allem Anschein nach die positiven und negativen Fragen im SUS-Fragebogen verwechselt wurden und damit eine verhältnismäßig niedrige SUS-Score entstanden wäre, obwohl die subjektive Frage zur Anwendung von den Teilnehmern als „Gut“ bis „Sehr gut“ bewertet wurde. Ein weiterer Teilnehmer der ersten Gruppe, der einen extremen negativen Ausreißer darstellte, wurde ebenfalls aus der Bewertung herausgenommen. Aus der letzten Gruppe wurde ebenfalls ein Teilnehmer aus der Bewertung ausgenommen, da auf Grund von sprachlichen Verständnisproblemen die Aufgaben zu einem großen Teil nicht gelöst werden konnten. Gesamt umfassen die Ergebnisse also 61 Teilnehmer, wobei 21 dem heterogenen OLAP-Tool zugeordnet werden können und 40 dem hetero-homogenen. Anzumerken ist noch, dass in der zweiten Gruppe kurzeitig der OLAP-Server ausgefallen war, da durch gleichzeitige Abfragen mit großem Volumen die Systemressourcen des Servers überlastet wurden, dies aber die Bewertung und den Arbeitsfluss der Probanden kaum störte da das Problem in kurzer Zeit gelöst werden konnte.

Zuerst werden die Daten des SUS-Fragebogens bzw. die damit direkt zusammenhängenden Daten gezeigt und danach jene die die einzelnen Tasks betreffen. Diagramm 1 zeigt wie viel Prozent der Probanden die Anwendung mit der jeweiligen SUS-Score bewertet haben. Nicht im Datensatz enthaltene SUS-Scores werden im Diagramm nicht gezeigt da ansonsten die Übersichtlichkeit leidet.

	Data Warehouse Typ			
	hetero-homogen		heterogen	
	Mittelwert	Median	Mittelwert	Median
Ich denke, dass ich dieses System gerne regelmäßig nutzen würde.	2,85	3,00	3,10	3,00
Ich fand das System unnötig komplex.	3,18	3,00	3,24	3,00
Ich denk, das System war leicht zu benutzen.	3,23	3,00	3,24	3,00
Ich denke, ich würde die Unterstützung einer fachkundigen Person benötigen, um das System benutzen zu können.	3,25	3,00	3,33	3,00
Ich fand, die verschiedenen Funktionen des Systems waren gut integriert.	3,03	3,00	3,05	3,00
Ich halte das System für inkonsistent.	3,38	4,00	3,43	4,00
Ich glaube, dass die meisten Menschen sehr schnell lernen würden, mit dem System umzugehen.	2,95	3,00	3,33	4,00
Ich fand das System sehr umständlich zu benutzen.	3,28	3,00	3,62	4,00
Ich fühlte mich bei der Nutzung des Systems sehr sicher.	2,78	3,00	3,05	3,00
Ich musste viele Dinge lernen, bevor ich mit dem System arbeiten konnte.	3,35	3,00	3,48	4,00
SUS-Score	**78,13**	**77,50**	**82,14**	**82,50**
Im gesamten würde ich die Nutzerfreundlichkeit dieser Anwendung folgendermaßen bewerten.	**5,50**	**6,00**	**5,62**	**6,00**

Tabelle 4: Mittelwert und Median der SUS-Scores und subjektive Bewertung

Tabelle 4 zeigt die Mittelwerte und Mediane der einzelnen Fragen des SUS-Fragebogens und der gesamten SUS-Score aufgeteilt in die Gruppen hetero-homogen und heterogen. Außerdem zeigt die Tabelle den Mittelwert und Median der subjektiven Frage zur Bewertung der Anwendung.

Diagramm 1 zeigt die Verteilung der SUS-Score der einzelnen Probanden und veranschaulicht wie die leichte Linksverschiebung des Mittelwertes bei dem hetero-homogenen OLAP-Tool im Vergleich zum heterogenen entsteht.

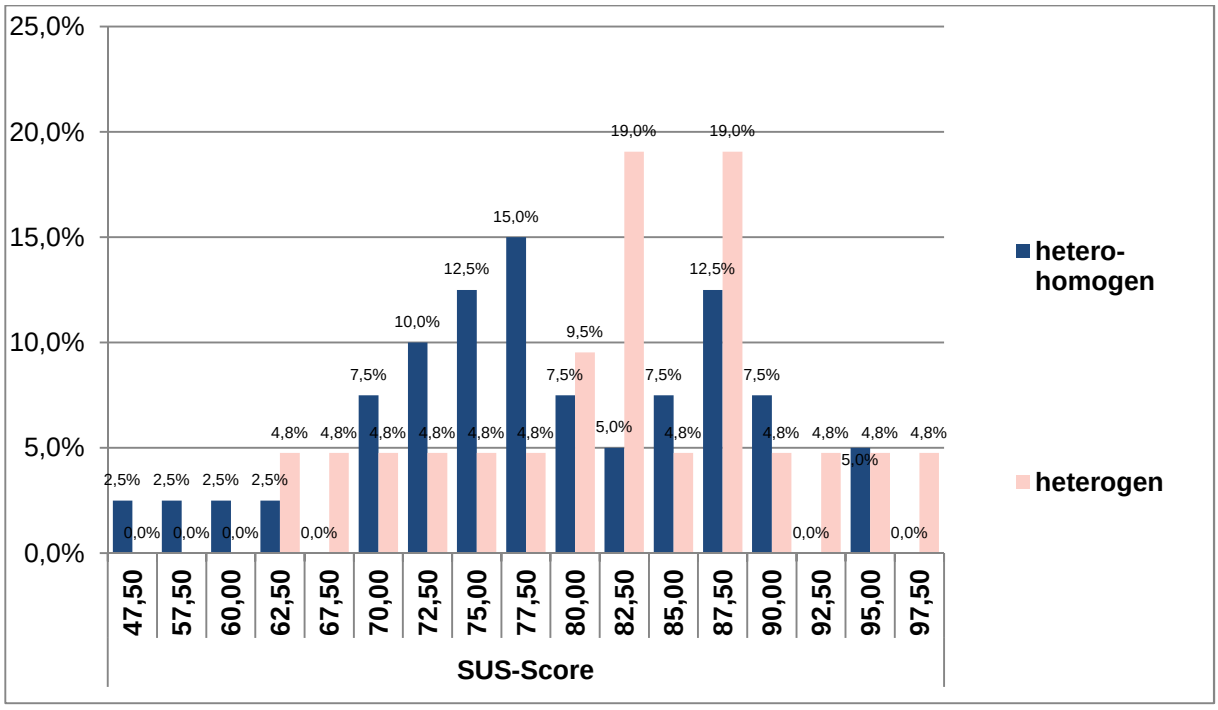


Diagramm 1: Verteilung der SUS-Scores

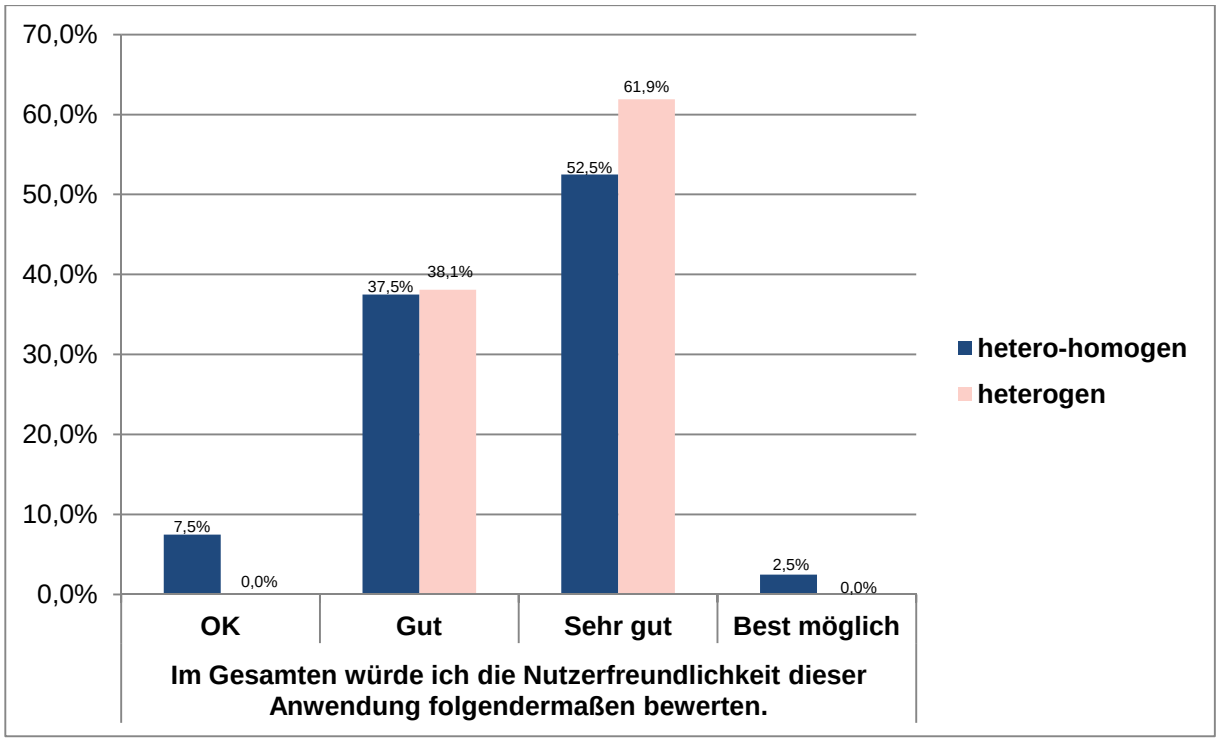


Diagramm 2: Auswertung subjektive Frage Nutzerfreundlichkeit der Anwendung

Ebenfalls zu sehen ist der gleiche Effekt bei der subjektiven Frage „Im Gesamten würde ich die Nutzerfreundlichkeit dieser Anwendung folgendermaßen bewerten“ wie Diagramm 2 zeigt.

Tabelle 5 zeigt als wie schwierig die Tasks im Durchschnitt von den Probanden für das jeweilige System empfunden wurden. Dabei zeigt sich, dass die Tasks 2-4 im Mittel und Median von den Probanden die das hetero-homogene OLAP-Tool verwenden mussten als etwas einfacher empfunden wurden und die Aufgaben 1 und 5 geringfügig schwieriger. Die Gründe dafür werden im nächsten Kapitel - in dem die Daten interpretiert und Schlüsse aus den Ergebnissen gezogen werden - besprochen.

	Data Warehouse Typ			
	hetero-homogen		heterogen	
	Mittelwert	Median	Mittelwert	Median
Task 1 Schwierigkeit	5,75	6,00	5,76	6,00
Task 2 Schwierigkeit	6,03	6,00	5,71	6,00
Task 3 Schwierigkeit	6,05	6,00	5,62	6,00
Task 4 Schwierigkeit	5,98	6,00	5,48	5,00
Task 5 Schwierigkeit	3,98	4,00	4,33	5,00

Tabelle 5: Mittelwert und Median der Schwierigkeit der einzelnen Tasks

Diagramm 3 bis Diagramm 12 zeigen wie schwierig die Probanden die Tasks anteilsmäßig empfunden haben und wie viele der Probanden anteilsmäßig den jeweiligen Task richtig, falsch oder nicht abgeschlossen haben. Diagramm 3 welcher Anteil an Probanden Task 1 richtig, falsch oder nicht abgeschlossen hat und Diagramm 4 zeigt wie die Beurteilung des Tasks von „sehr leicht“ bis „sehr schwierig“ ausgefallen ist. Dabei stellt sich heraus, dass jene Probanden die das hetero-homogene OLAP-Tool verwendeten bei diesem Task zwar geringfügig mehr Fehler und unvollständige Antworten gaben aber die Probanden die richtige Antworten gaben bewerteten das hetero-homogene OLAP-Tool durchschnittlich als leichter als die Probanden des heterogenen OLAP-Tools.

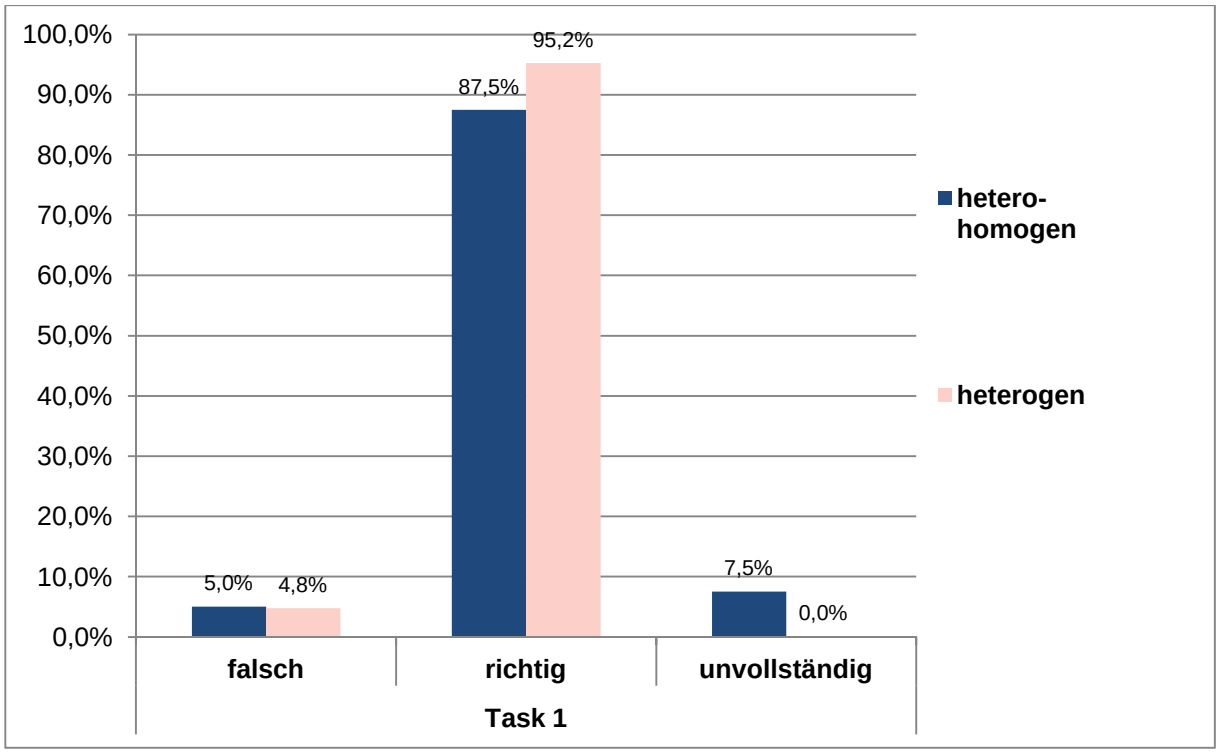


Diagramm 3: Task 1 abgeschlossen

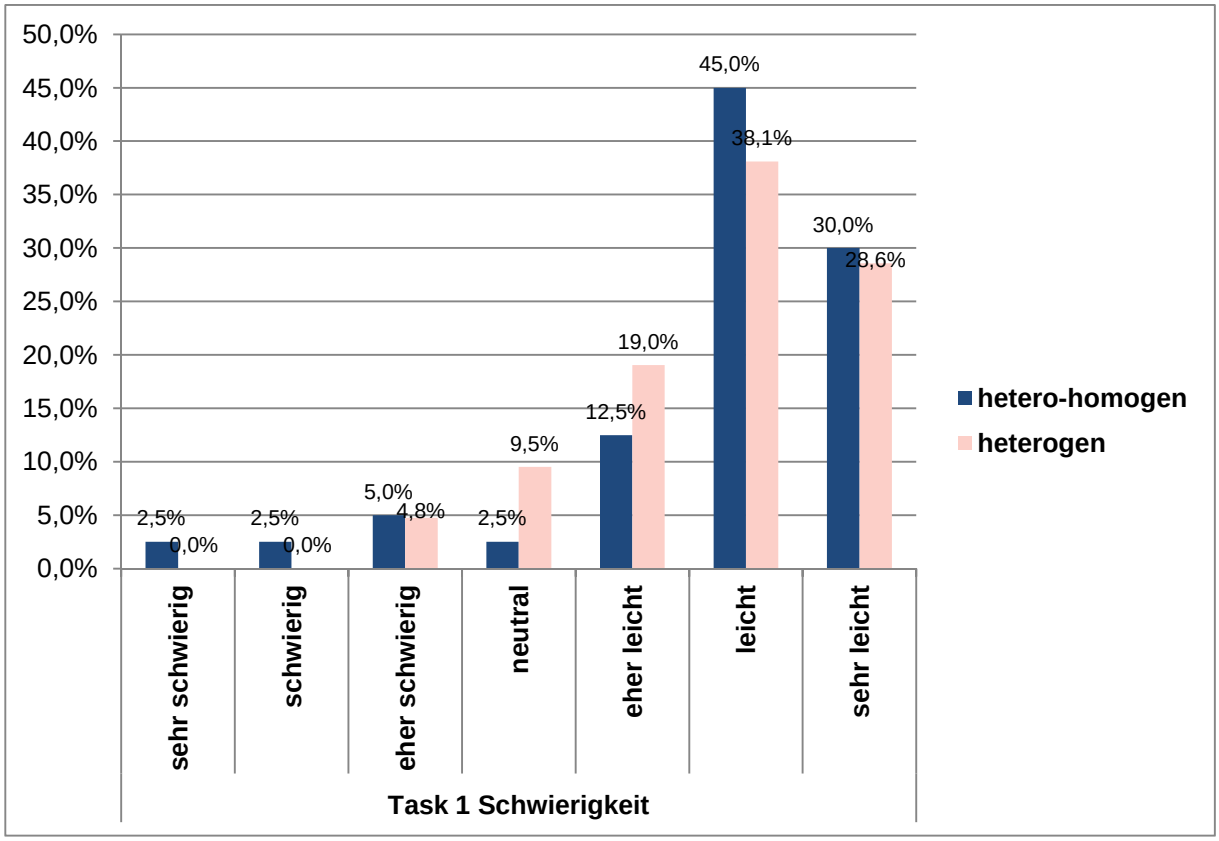


Diagramm 4: Task 1 Schwierigkeit

Diagramm 5 zeigt, dass bei Task 2 ein etwas größerer Anteil an Probanden mit dem heterogenen OLAP-Tool Fehler produzierte und Diagramm 6 zeigt, dass dem entsprechend ein deutlich größerer Anteil an Probanden, die mit dem hetero-homogenen OLAP-Tool arbeiteten diesen Task als leicht oder sehr leicht empfunden haben.

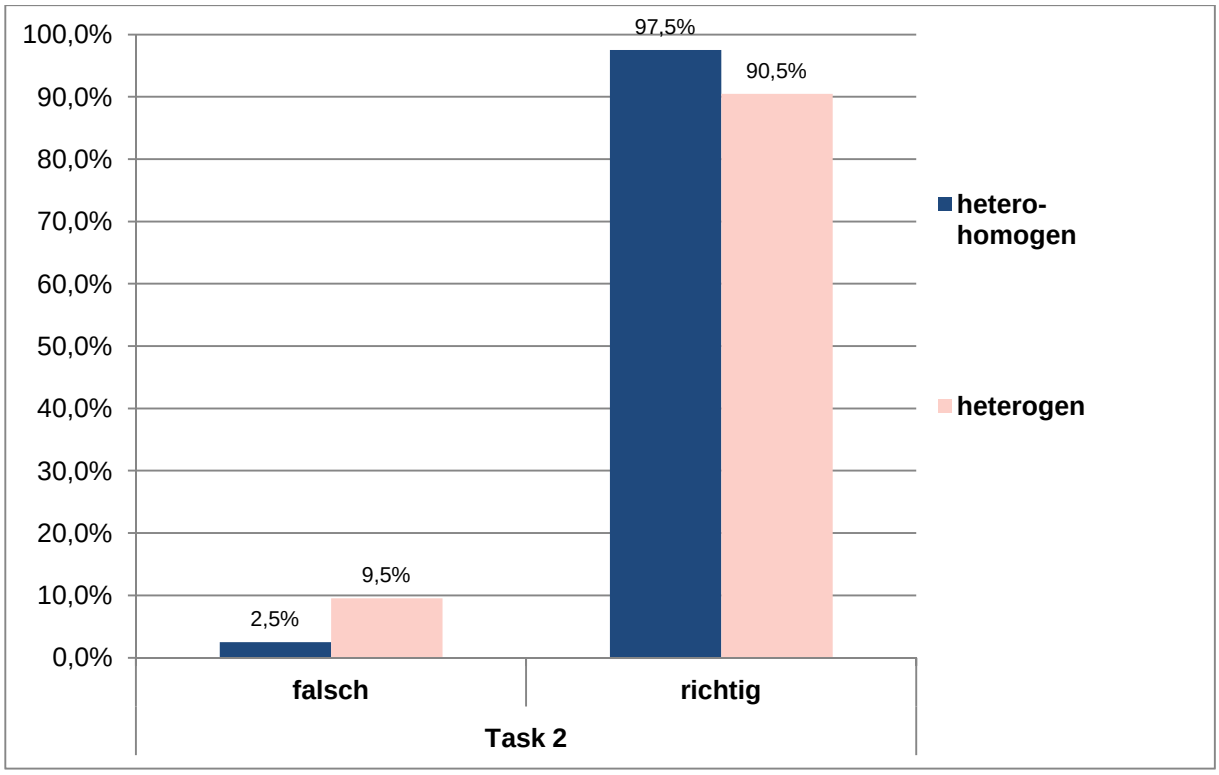


Diagramm 5: Task 2 abgeschlossen

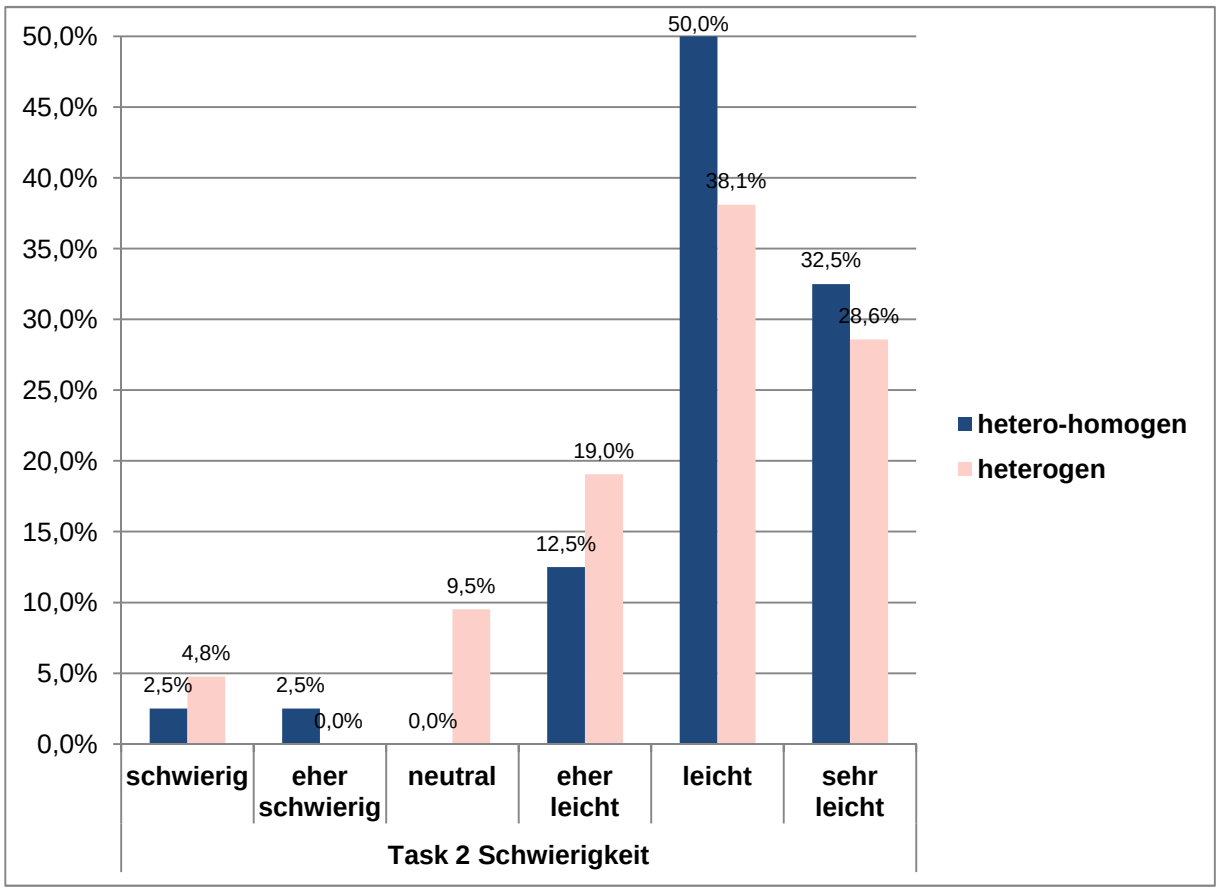


Diagramm 6: Task 2 Schwierigkeit

Die Ergebnisse zu Task 3 liefern wie in Diagramm 7 und Diagramm 8 zu sehen ist sehr ähnliche Werte wie Task 2.

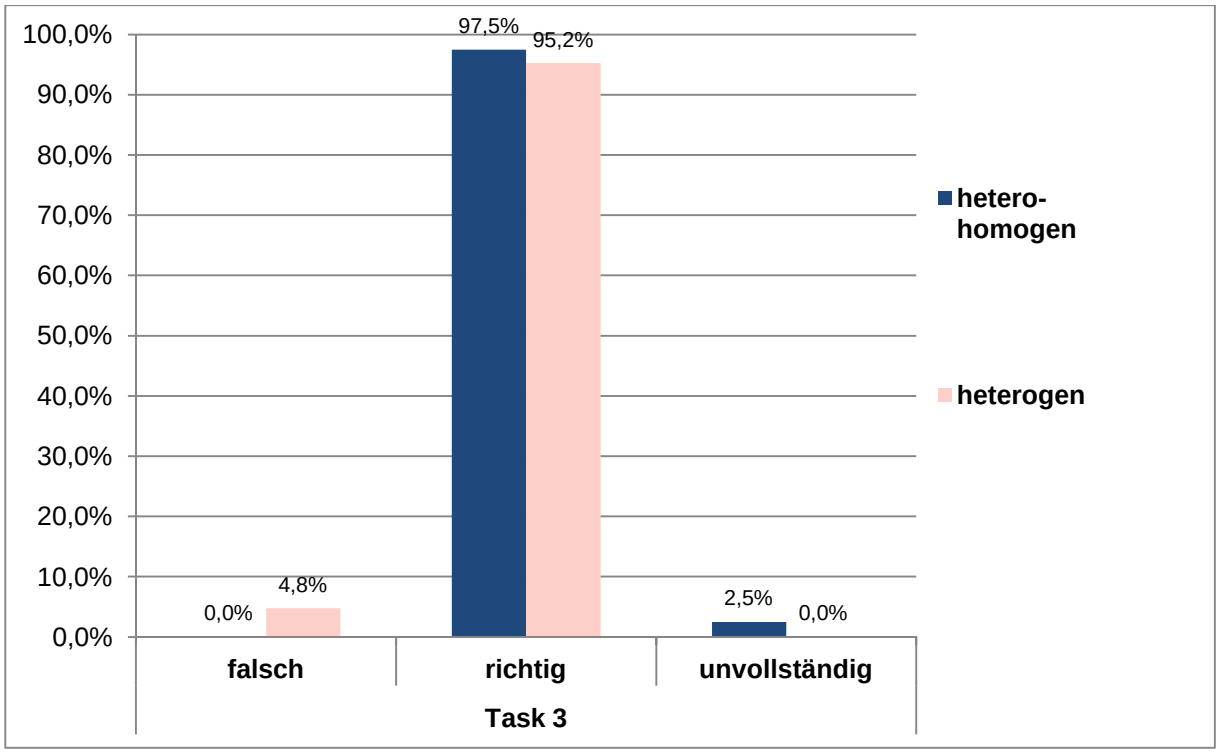


Diagramm 7: Task 3 abgeschlossen

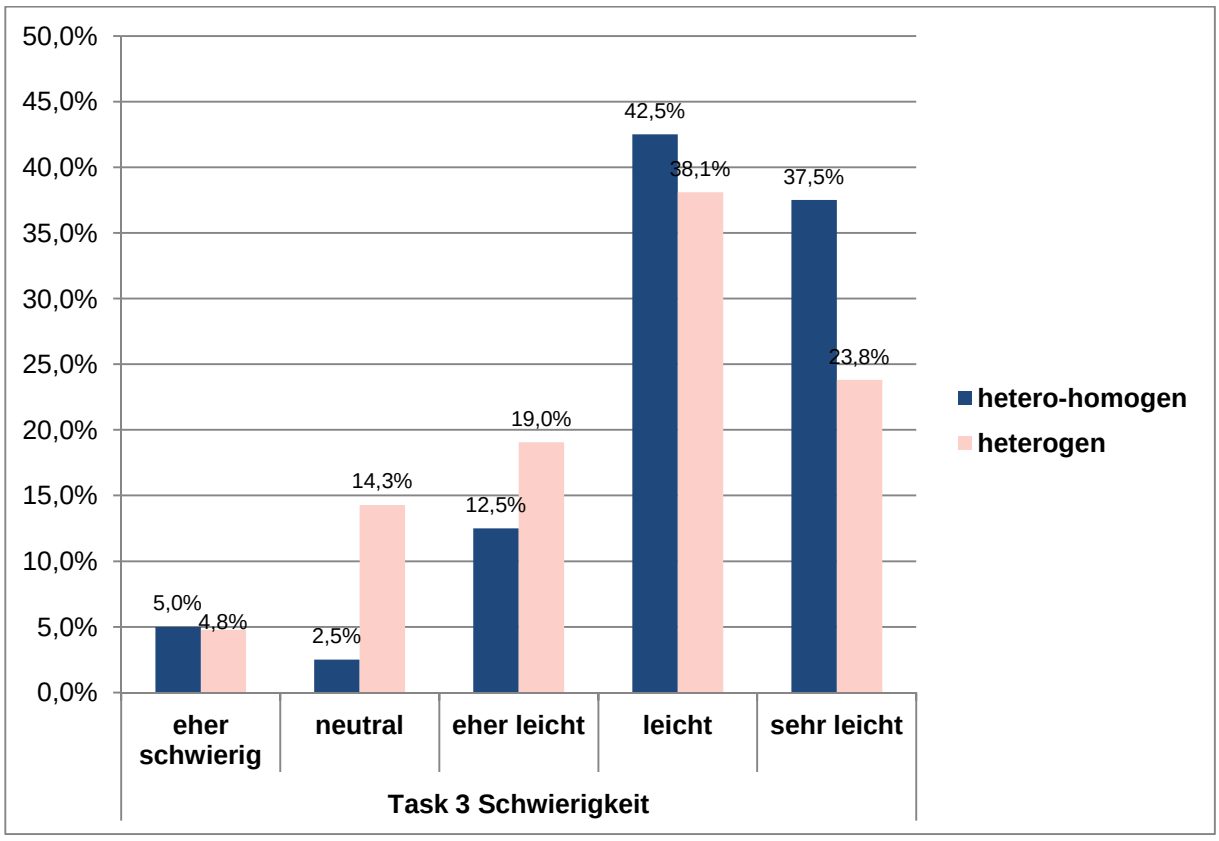


Diagramm 8: Task 3 Schwierigkeit

Die Ergebnisse aus Task 4 (Diagramm 9 und Diagramm 10) zeigen erneut wie in Task 1, dass zwar ein geringfügig größerer Anteil an Probanden die das hetero-homogene OLAP-Tool verwendeten etwas mehr Fehler machten bzw. den Task unvollständig beendeten aber die Aufgabe in Summe als einfacher bewertet wurde als von den Probanden des heterogenen OLAP-Tools.

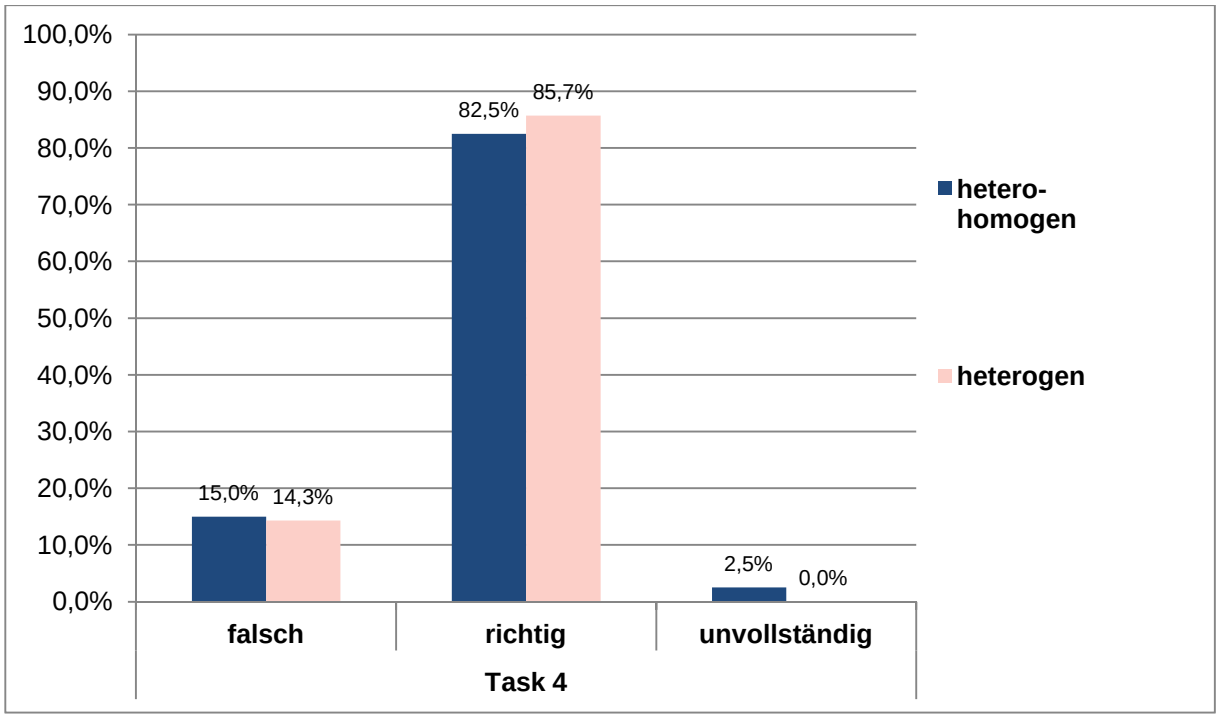


Diagramm 9: Task 4 abgeschlossen

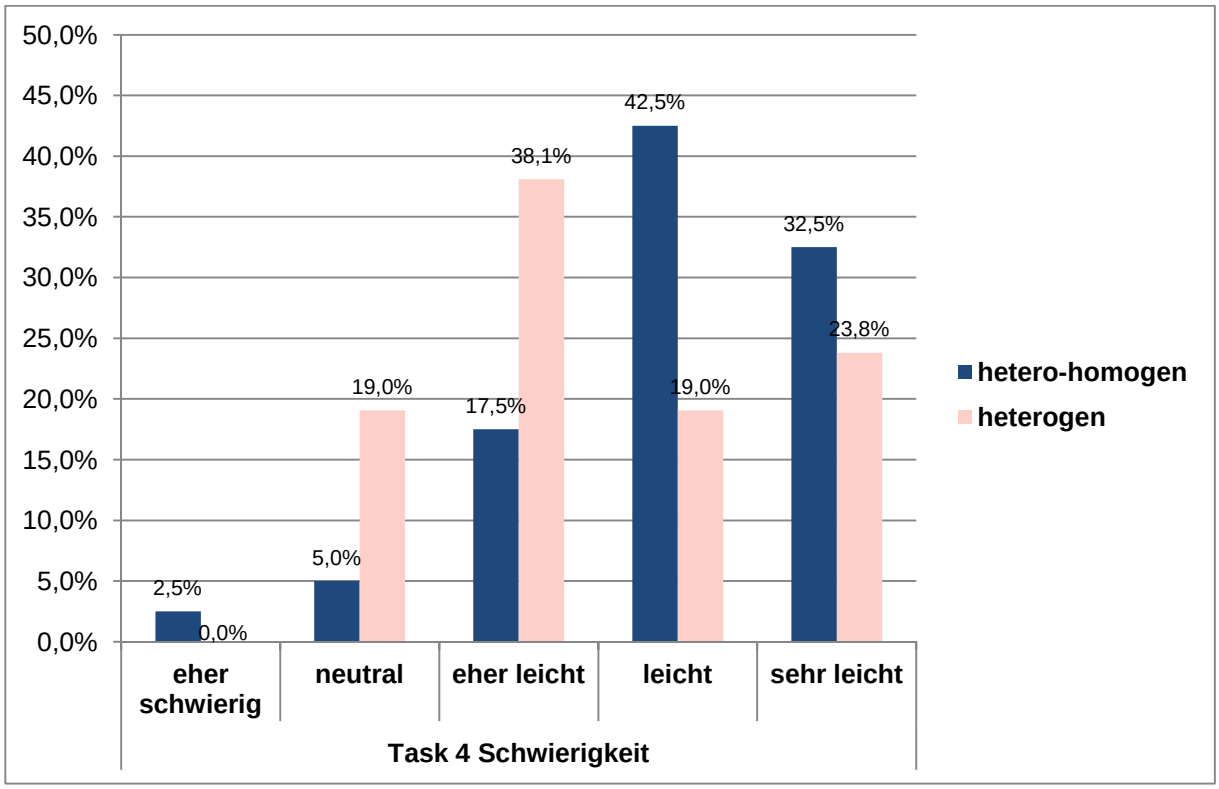


Diagramm 10: Task 4 Schwierigkeit

Diagramm 11 und Diagramm 12 zeigen, dass bei Task 5 ein geringfügig größerer Anteil der Probanden die mit dem hetero-homogenen OLAP-Tool arbeiteten Fehler in der Auswertung machten und ein deutlich größerer Anteil diesen Task nicht vollständig beendete, dementsprechend wurde dieser Task auch von den Probanden des heterogenen OLAP-Tools als einfacher empfunden.

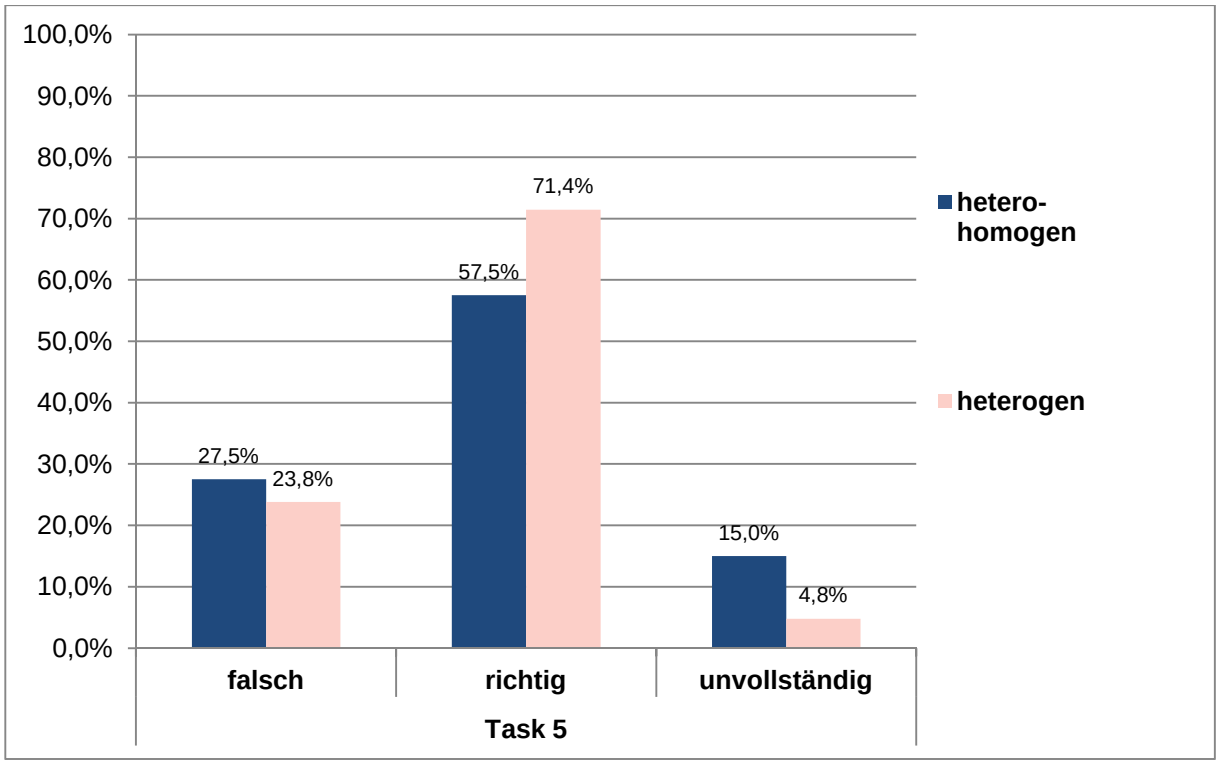


Diagramm 11: Task 5 abgeschlossen

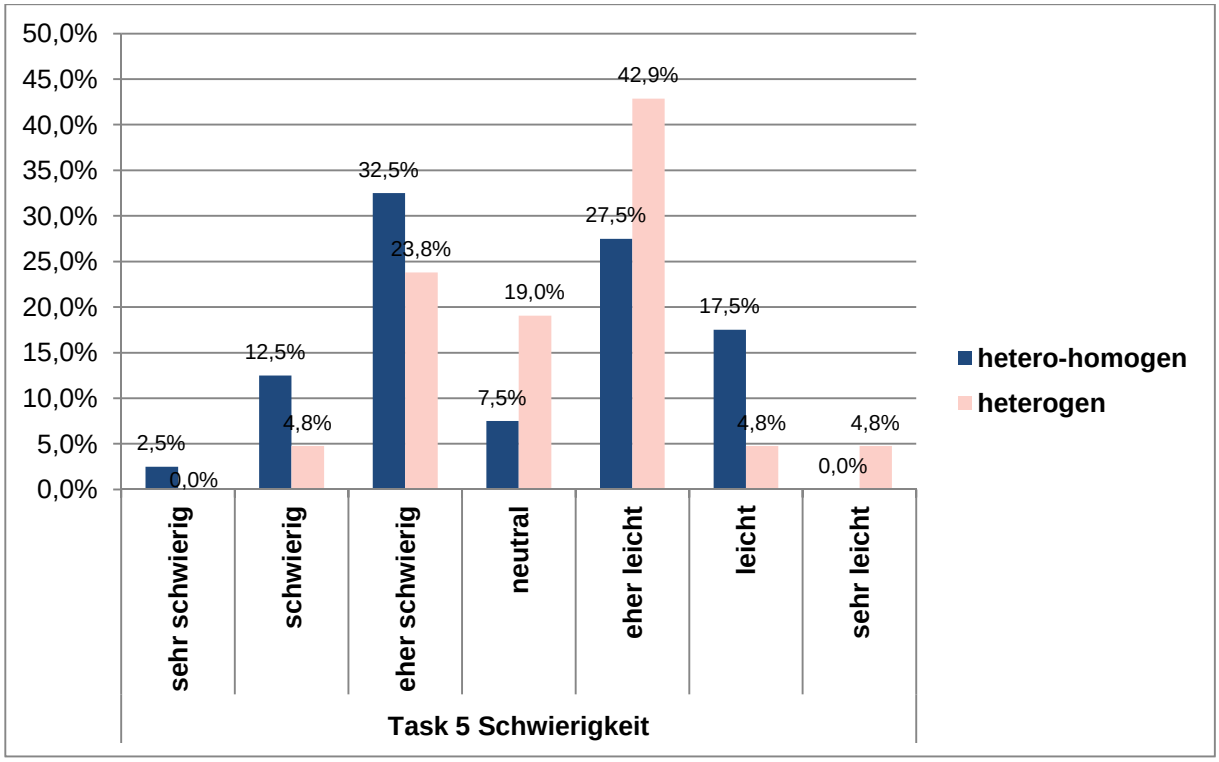


Diagramm 12: Task 5 Schwierigkeit

5.3. Interpretation

In dieser Studie geht es in erster Linie darum festzustellen wie der hetero-homogene Ansatz von Anwendern in Bezug auf die Usability angenommen wird, da auf Grund der Bedingungen in der Suche selbst, zu Gunsten von besserer Datenqualität ohne Informationsverlust, diese möglicherweise leidet.

Laut Bangor, Kortum und Miller liegt die durchschnittliche SUS-Score (Mittelwert) bei internetbasierten Webseiten bzw. Applikationen aus den erhobenen Daten von 1180 Fragebögen bei 68,05 (vgl. [17] S. 583), dieser Wert wird bei der hier gemachten Studie für den hetero-homogenen Ansatz unter der Nutzung des OLAP-Tools Saiku 2.4 deutlich mit einem Mittelwert von 78,13 bzw. einem Median von 77,50 bei 40 Probanden überschritten und zeigt damit, dass dieser Ansatz ein hohes Potenzial hat. Vergleicht man die SUS-Score des hetero-homogenen OLAP-Tools aber mit der des rein heterogenen (siehe Abbildung 23), zeigt sich, dass die Usability hier durch die Probanden (n=21) mit einer SUS-Score im Mittel von 82,14 bzw. mit einem Median von 82,50 für das heterogene OLAP-Tool geringfügig besser bewertet wird sich aber beide im Bereich „Gut“ befinden.

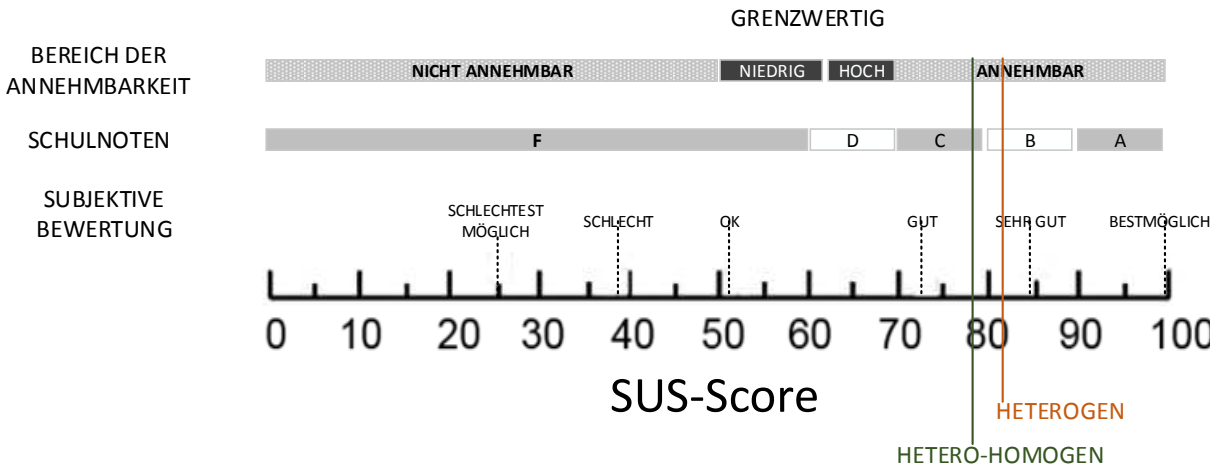
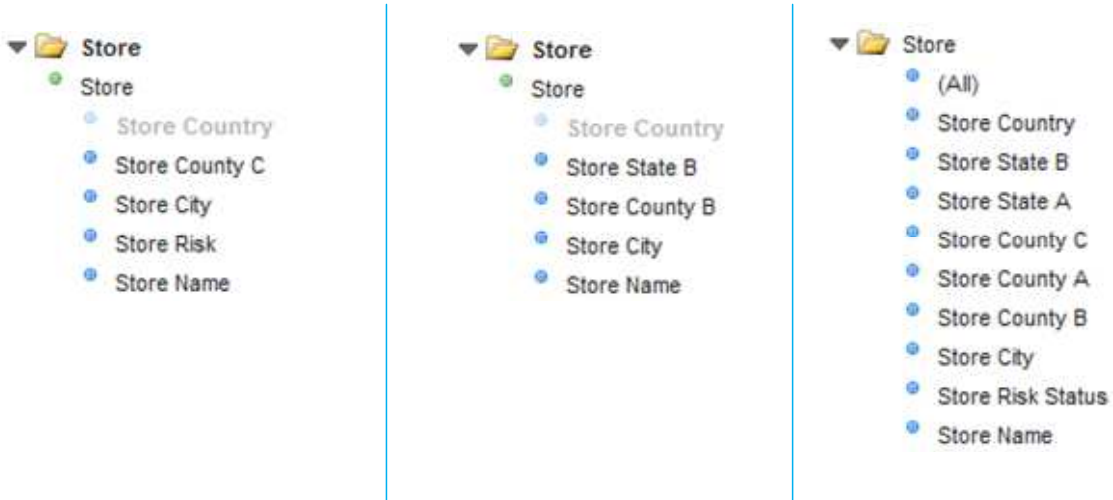


Abbildung 22: SUS-Score Vergleich hetero-homogen zu heterogen [25]

Betrachtet man die einzelnen 10 Fragen des SUS-Fragebogens zeigt sich, dass der größte Unterschied im Mittel mit 0,38 und 0,34 bei den beiden Fragen „Ich glaube, dass die meisten Menschen sehr schnell lernen würden, mit dem System umzugehen“ und „Ich fand das System sehr umständlich zu benutzen“ besteht. Multipliziert man diese beiden Werte mit 2,5, wie es zur Ermittlung der SUS-Score notwendig ist, entsteht dadurch ein Unterschied von 1,8 Punkten. Dies ist besonders interessant, da ein Proband der das hetero-homogene OLAP-Tool verwendete zu der Frage „Ich glaube, dass die meisten Menschen sehr schnell lernen würden, mit dem System umzugehen“ den Satz „Sofern man damit zu tun hat und die Zahlen und die Ergebnisse versteht“ hinzufügte. Diese Aussage zusammen mit dem Ergebnis der Fehlerquote und Anzahl der nicht abgeschlossenen Tasks und der Befragung zur Schwierigkeit der Tasks die bis auf den ersten (wobei dieser von beiden Gruppen als quasi gleich schwierig betrachtet wurde) und den letzten Task von den Probanden des hetero-homogenen OLAP-Tools als einfacher empfunden wurden zeigt, dass vermutlich in erster Linie ein fehlendes Wissen zur genauen Funktionalität (diese wurde zwar in der Einführung gezeigt aber anscheinend von vielen schnell vergessen) des hetero-homogenen Ansatzes zu dem geringfügig schlechterem Ergebnis geführt hat. Denn in genau diesen beiden Tasks wurde nach einer Kennzahl gefragt die im hetero-homogenen OLAP-Tool nur verfügbar ist wenn sich der Anwender mit seiner Filterauswahl in einem Bereich des Datenwürfels befindet der auch tatsächlich Werte für diese Kennzahl besitzt und dieser bei falscher oder nicht durchgeführten Filterung nicht sichtbar ist und damit eine Frustration bei dem Anwender auslösen kann wenn er diesen nicht in angemessener Zeit findet.

Es hat sich auch bei den Tasks 2-4 gezeigt, dass die Probanden die das heterogene OLAP-Tool verwendeten mehr Aktionen tätigen mussten um die richtigen Daten zu finden, da in dieser Anwendung die Hierarchien in den Dimensionen deutlich mehr Ebenen haben, die mitunter sehr ähnliche Namen tragen. Abbildung 23 zeigt wie die „Store“-Hierarchien im hetero-homogenen OLAP-Tool in Task 2 und 4 ausgesehen haben wenn der Filter in „Store Country“ der Aufgabenstellung entsprechend richtig gesetzt wurde und wie die „Store“-Hierarchie im heterogenen OLAP-Tool immer aussieht.

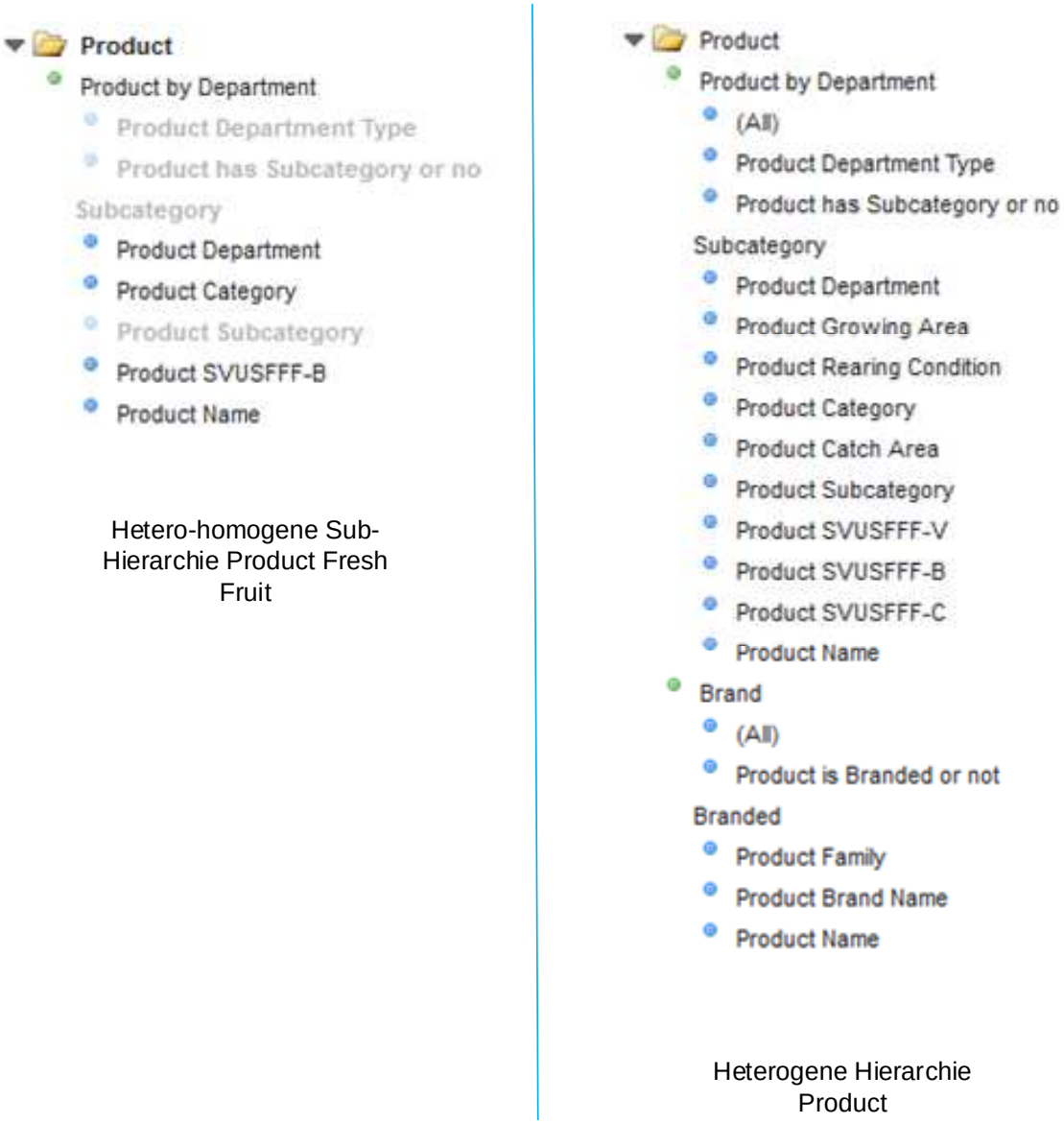


Hetero-homogene Sub-Hierarchie Store Mexico

Hetero-homogene Sub-Hierarchie Store Canada

Heterogene Hierarchie Store

Abbildung 23: Hetero-homogene Hierarchien und heterogene Hierarchie Store



Hetero-homogene Sub-Hierarchie Product Fresh Fruit

Heterogene Hierarchie Product

Abbildung 24: Hetero-homogene und heterogene Hierarchie „Product“

In Task 3 war der Profit für Produkte vom „Product Department Type“ „Multicategory“ gefragt und im weiteren Verlauf von Produkten mit Subkategorien und dann der Subkategorie „Fresh Fruit“ und wie dieser über die „SVUSFFF“-Ziffer aufgeteilt ist. Abbildung 24 zeigt wie die Hierarchie bei richtiger Filterung im hetero-homogenen OLAP-Tool aussieht und wie im heterogenen OLAP-Tool.

Gerade bei diesem Task mussten die Probanden die das heterogene OLAP-Tool verwendeten viele Aktionen durchführen um die richtige „SVUSFFF“ Ebene zu finden. Der Ausschnitt (Tabelle 6) aus Piwik zeigt wie der Großteil der Probanden im heterogenen OLAP-Tool vorgegangen ist um diesen Task zu lösen. Es ist zu sehen, dass im Endeffekt alle drei Ebenen in der Hierarchie die eine „SVUSFFF“-Bezeichnung tragen verwendet wurden da man nicht wusste welche hier gültig ist.

Schritt	Proband	Aktion
94.	hetdwh11	Level - ondrag - [Product.Product by Department].[Product Department Type]
95.	hetdwh11	Filter Dialog für - [Product.Product by Department].[Product Department Type] geöffnet
96.	hetdwh11	Clicked on - add members
97.	hetdwh11	Clicked on - save
98.	hetdwh11	Measures - ondrag - [Measures].[Profit]
99.	hetdwh11	moved #Product/hierarchy/[Product.Product by Department]/[Product.Product by Department].[Product Department Type] to - COLUMNS
100.	hetdwh11	moved #Product/hierarchy/[Product.Product by Department]/[Product.Product by Department].[Product Department Type] to - ROWS
101.	hetdwh11	Filter Dialog für - [Product.Product by Department].[Product Department Type] geöffnet
102.	hetdwh11	Run Query - clicked
103.	hetdwh11	Level - ondrag - [Product.Product by Department].[Product has Subcategory or no Subcategory]
104.	hetdwh11	Filter Dialog für - [Product.Product by Department].[Product has Subcategory or no Subcategory] geöffnet
105.	hetdwh11	Clicked on - add members
106.	hetdwh11	Clicked on - save
107.	hetdwh11	Run Query - clicked
108.	hetdwh11	Filter Dialog für - [Product.Product by Department].[Product has Subcategory or no Subcategory] geöffnet
109.	hetdwh11	Level - ondrag - [Product.Product by Department].[Product Subcategory]
110.	hetdwh11	Filter Dialog für - [Product.Product by Department].[Product Subcategory] geöffnet
111.	hetdwh11	Clicked on - add members
112.	hetdwh11	Clicked on - save
113.	hetdwh11	Run Query - clicked
114.	hetdwh11	**Level - ondrag - [Product.Product by Department].[Product SVUSFFF-V]**
115.	hetdwh11	Run Query - clicked
116.	hetdwh11	**removed #Product/hierarchy/[Product.Product by Department]/[Product.Product by Department].[Product SVUSFFF-V]**
117.	hetdwh11	**Level - ondrag - [Product.Product by Department].[Product SVUSFFF-B]**
118.	hetdwh11	**removed #Product/hierarchy/[Product.Product by Department]/[Product.Product by Department].[Product SVUSFFF-B]**
119.	hetdwh11	**Level - ondrag - [Product.Product by Department].[Product SVUSFFF-C]**
120.	hetdwh11	Run Query - clicked
121.	hetdwh11	Filter Dialog für - [Product.Product by Department].[Product Subcategory] geöffnet
122.	hetdwh11	**Filter Dialog für - [Product.Product by Department].[Product SVUSFFF-C] geöffnet**
123.	hetdwh11	**removed #Product/hierarchy/[Product.Product by Department]/[Product.Product by Department].[Product SVUSFFF-C]**
124.	hetdwh11	**Level - ondrag - [Product.Product by Department].[Product SVUSFFF-B]**
125.	hetdwh11	**Filter Dialog für - [Product.Product by Department].[Product SVUSFFF-B] geöffnet**
126.	hetdwh11	Clicked on - close
127.	hetdwh11	Run Query - clicked
128.	hetdwh11	Task abgeschlossen

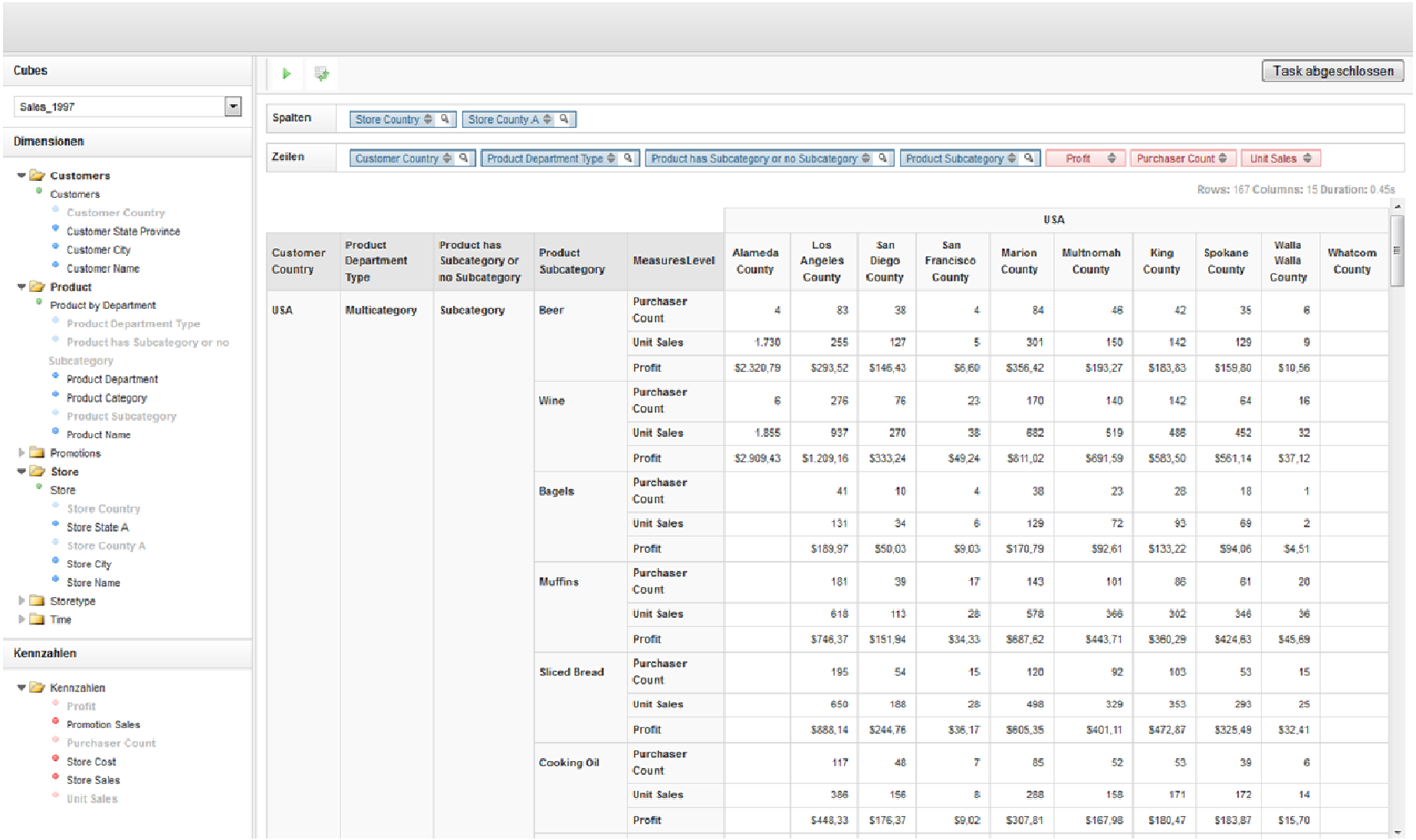
Tabelle 6: Aktionen eines Probanden zu Task 3 in heterogenen OLAP-Tool

In den Diagrammen in Kapitel 5.2 zu Task 4 (Diagramm 9) und Task 5 (Diagramm 11) ist deutlich zu sehen, dass hier mehr Fehler, sowohl in den hetero-homogenen Gruppen als auch der heterogenen Gruppe, vorkamen. Dies liegt vermutlich vor allem daran, dass bei diesen beiden Tasks keine weiteren Informationen mehr gegeben wurden welche Ebenen benötigt werden und welche Filter auf diesen gesetzt werden müssen, was in den Tasks 1-3 noch teilweise geschah und sukzessive abgebaut wurde um langsam freieres Handeln zu erzwingen. In Task 4 entstanden so die meisten Fehler weil die Probanden vergessen oder überlesen haben, dass nur Produkte die eine Marke besitzen, gesucht sind. In Task 5 wurden von den Gruppen noch mehr Fehler gemacht, da hier deutlich mehr Ebenen und eine genauere Sicht auf Daten und Filter gelegt werden musste. Wurden hier beispielsweise die Daten nicht gefiltert waren die Probanden mit einer Informationsüberflutung konfrontiert. Abbildung 25 zeigt welche Masse an Informationen die Probanden erwartete, wenn nicht genau nach den gefragten Informationen gefiltert wurde (dabei ist in dem Screenshot nur ein sehr kleiner Teil der Daten zu sehen, wie die Bildlaufleiste zeigt).

Außerdem wurde von sehr vielen Probanden, die das hetero-homogene OLAP-Tool verwendeten, nicht sowohl auf amerikanische Kunden und amerikanische Geschäfte gefiltert sondern lediglich auf amerikanische Geschäfte, wodurch für diese die Kennzahl „Purchaser Count“ nicht angezeigt wurde und somit der Task nicht abgeschlossen werden konnte. Diesen Fehler machten laut Auswertung des Logs auch 7 der 21 Probanden im heterogenen OLAP-Tool. Da in den Daten tatsächlich nur amerikanische Kunden in amerikanischen Geschäften Umsätze generiert haben stimmen in diesem Fall zwar die Ergebnisse ohne Filter auf Kunden, allerdings hätte diese Vorgehensweise zu Fehlern geführt wenn Umsätze durch Kunden anderer Länder in den USA gemacht worden wären, diese Art von Fehler ist durch den hetero-homogenen Ansatz ausgeschlossen.

Fazit

Die Studie zeigt, dass der hetero-homogene Ansatz eine gute Usability besitzt und ohne große Vorkenntnisse des Anwenders bis zu einem gewissen Grad auch einfach verwendbar ist. Allerdings hat sich gezeigt, dass bei komplexeren Abfragen mit vielen Bedingungen, und im Besonderen verschiedenen Kennzahlen die nur in Teilbereichen eines Würfels vorhanden sind, ohne jegliches Hintergrundwissen zur Datenlage und Datenstrukturierung mitunter Probleme auftreten können. Für einfachere Abfragen bieten die Restriktionen in der Datenauswahl aber auch Vorteile wie sich bei der Auswertung zur Schwierigkeit der Tasks 2-4 gezeigt hat. Um sicherzustellen, dass Anwender auch ohne fundierte Kenntnisse zur Datenlage und ihrem internen Aufbau in einem Data Warehouse mit hetero-homogener Modellierung gut umgehen können, wäre es notwendig diese entweder durch Visualisierung oder andere adäquate Methoden direkt in dem verwendeten OLAP-Tool sichtbar zu machen um Missverständnisse zu vermeiden und die Verwendung zu vereinfachen.



					USA									
Customer Country	Product Department Type	Product has Subcategory or no Subcategory	Product Subcategory	MeasuresLevel	Alameda County	Los Angeles County	San Diego County	San Francisco County	Marion County	Multnomah County	King County	Spokane County	Walla Walla County	Whatcom County
USA	Multicategory	Subcategory	Beer	Purchaser Count	4	83	38	4	84	46	42	35	6	
				Unit Sales	1.730	255	127	5	301	150	142	129	9	
				Profit	$2.320,79	$293,52	$146,43	$6,60	$356,42	$193,27	$183,83	$159,80	$10,56	
			Wine	Purchaser Count	6	276	76	23	170	140	142	64	16	
				Unit Sales	1.855	937	270	38	682	519	488	452	32	
				Profit	$2.909,43	$1.209,16	$333,24	$49,24	$811,02	$691,59	$583,50	$561,14	$37,12	
			Bagels	Purchaser Count		41	10	4	38	23	28	18	1	
				Unit Sales		131	34	6	129	72	93	69	2	
				Profit		$189,97	$50,03	$9,03	$170,79	$92,61	$133,22	$94,06	$4,51	
			Muffins	Purchaser Count		181	39	17	143	101	86	61	20	
				Unit Sales		618	113	28	578	366	302	346	36	
				Profit		$746,37	$151,94	$34,33	$687,62	$443,71	$360,29	$424,63	$45,69	
			Sliced Bread	Purchaser Count		195	54	15	120	92	103	53	15	
				Unit Sales		650	188	28	498	329	353	293	25	
				Profit		$888,14	$244,76	$36,17	$605,35	$401,11	$472,87	$325,49	$32,41	
			Cooking Oil	Purchaser Count		117	48	7	85	52	53	39	6	
				Unit Sales		388	156	8	288	158	171	172	14	
				Profit		$448,33	$176,37	$9,02	$307,81	$167,98	$180,47	$183,87	$15,70	

Abbildung 25: Abfrageergebnis in Task 5 ohne Filterung

6. Zusammenfassung und Ausblick

In dieser Arbeit wurde einerseits unter Verwendung des Open-Source OLAP-Tools Saiku in der Version 2.4 in Verbindung mit Mondrian 3 und einem bestehendem Prototypen in Oracle der als Index verwendet wurde gezeigt wie hetero-homogene Data Warehouses in relationalen Datenbanksystemen abgebildet und genutzt werden können und andererseits wurde, anhand einer Benutzerstudie geprüft ob dieser Ansatz für OLAP in hetero-homogenen Data Warehouses einen brauchbaren Grad an Usability gewährleistet. Dazu wurden ein für diese Zwecke etablierter Fragebogen, der sich System Usability Scale nennt, gemeinsam mit einer post-Task Frage zur Ermittlung der Schwierigkeit einzelner Aufgaben verwendet die von 40 Probanden nach Einsatz des OLAP-Tools mit hetero-homogener Umsetzung und 21 Probanden nach Einsatz des gleichen OLAP-Tools mit rein heterogener Umsetzung beantwortet wurden. Das Ergebnis dieser Studie hat gezeigt, dass der hetero-homogene Ansatz prinzipiell eine gute bis sehr gute Usability besitzt und bei bestimmten Aufgaben durch die eindeutige Zuordnung von heterogenen Ebenen in Dimensionshierarchien und heterogenen Kennzahlen die Auswahl vereinfacht, andererseits hat sich auch gezeigt, dass genau dieser Effekt bei komplexeren Abfragen in denen viele heterogene Dimensionen und Kennzahlen gleichzeitig für eine Abfrage benötigt werden zu Problemen führen kann wenn der Anwender kein oder wenig Wissen zur Datenstruktur bzw. Datenlage besitzt da hier schnell eine gewisse Frustration entstehen kann wenn auf Grund der gegeben Restriktionen Ebenen und Kennzahlen nicht für den Anwender sichtbar sind die er eigentlich erwarten würde.

Auf Grund dieses Ergebnisses scheint es notwendig eine Art Wegweiser für den Anwender in einem OLAP-Tool, das den hetero-homogenen Ansatz umsetzt, bereitzustellen um sicher gehen zu können, dass sich der Anwender im hetero-homogenen Raum zurechtfindet und auch ohne fundiertes Hintergrundwissen mit der Anwendung umgehen kann. Des Weiteren wurde in dieser Arbeit, wie erwähnt, ein bereits bestehender Prototyp verwendet um die hetero-homogene Funktionalität abbilden zu können. Aus Gründen der Effizienz wäre es hier notwendig eine eigenständige Lösung zu implementieren die diesen nicht mehr benötigt.

A Appendix

Installation

Der hier beschriebene Installations- und Konfigurationsvorgang zeigt wie die Data Warehouses und ihre Komponenten für die Benutzerstudie aufgebaut werden. Prinzipiell sind aber mit den Anwendungen beliebige logische und physische Schemata verwendbar, für das hetero-homogene Data Warehouse muss sich aber bei dem Aufbau des Mondrian-Schemas an bestimmte Namenskonventionen gehalten werden die bereits in Kapitel 4.2.2 dieser Arbeit besprochen wurden. Die beiden OLAP-Tools mit den Mondrian-Schemata, SQL-Dumpfiles und Scripts für Oracle finden Sie auf der Webseite des HH-DW Projects[1].

Allgemeine Installation und Konfigurationen

Installieren Sie MySQL in der Version 5.1.73[2].

Öffnen Sie das Dumpfile foodmart_copy.sql mit einem beliebigen Editor und suchen Sie nach DEFINER=``@``, dies muss ersetzt werden durch DEFINER=`username`@`localhost`. username steht hierbei für den Benutzer den Sie in ihrer MySQL Datenbank angelegt haben, dieser muss Administratorrechte besitzen. Laden Sie anschließend foodmart_copy.sql in MySQL, es sollten die Datenbank mit Tabellen und Views erzeugt werden.

Installieren Sie falls nicht vorhanden Java in der Version 1.7 64 bit[3] und setzen sie das JDK auf die Umgebungsvariable JAVA_HOME.

Installation und Konfiguration heterogenes Data Warehouse

Um Saiku 2.4 mit dem heterogenen Setup nutzen zu können müssen Sie die ZIP-Datei Saiku-2.4-heterogen entpacken und in dem Ordner ../Saiku-2.4-heterogen/saiku-server/tomcat/webapps/saiku/WEB-INF/classes/saiku-datasources/ die Datei foodmart_test mit einem beliebigen Editor öffnen und bei username= und password= den Benutzernamen und das Passwort für die foodmart_copy Datenbank die Sie zuvor erstellt haben angeben.

Des Weiteren müssen Sie einen MySQL Connector/J in der Version 5.1.*[4] in den Ordner ../Saiku-2.4-heterogen/saiku-server/tomcat/webapps/saiku/WEB-INF/lib kopieren um sich mit der MySQL Datenbank verbinden zu können.

Die Mondrian-Schemadatei ist unter ../Saiku-2.4-heterogen/saiku-server/tomcat/webapps/saiku/WEB-INF/classes/foodmart_test zu finden.

Installation und Konfiguration hetero-homogenes Data Warehouse

Um Saiku 2.4 mit dem hetero-homogenen Setup nutzen zu können müssen Sie zuerst die Oracle 11g R2 Datenbank oder eine neuere Version installieren[5]. Installationsanweisungen zu Oracle 11g Release 2 finden sie in der Oracle Dokumentation[6].

Nach der Installation von Oracle müssen Sie HH-DW (The Hetero-Homogeneous Data Warehouse Project) darauf installieren um das Datenbanksystem um hetero-homogene Funktionalitäten zu erweitern. Genauere Angaben dazu finden sie in der README Datei dieses Projekts.

[1] hh-dw.dke.uni-linz.ac.at
[2] https://dev.mysql.com/downloads/mysql/5.1.html
[3] http://www.oracle.com/technetwork/java/javase/downloads/jdk7-downloads-1880260.html
[4] https://dev.mysql.com/downloads/connector/j/
[5] http://www.oracle.com/technetwork/database/enterprise-edition/downloads/112010-linx8664soft-100572.html
[6] http://docs.oracle.com/cd/E11882_01/nav/portal_11.htm

Nachdem Sie HH-DW erfolgreich installiert haben führen Sie die beiden Skripte dimensions.sql und cubes.sql in dieser Reihenfolge auf der Oracle-Datenbank aus um die notwendigen M-Objects und M-Relationships auf der Oracle-Datenbank zu erzeugen.

Entpacken Sie jetzt die ZIP-Datei Saiku-2.4-hetero-homogen und öffnen Sie mit einem beliebigen Editor die Datei foodmart_test_hh im Ordner ../Saiku-2.4-hetero-homogen/saiku-server/tomcat/webapps/saiku/WEB-INF/classes/saiku-datasources/. In dieser Datei müssen Sie bei username= und password= den Benutzernamen und das Passwort für die foodmart_copy Datenbank die Sie zuvor erstellt haben angeben.

Des Weiteren müssen Sie einen MySQL Connector/J in der Version 5.1.* [7] in den Ordner ../Saiku-2.4-hetero-homogen/saiku-server/tomcat/webapps/saiku/WEB-INF/lib kopieren um sich mit der MySQL Datenbank verbinden zu können.

Um den Webservice der sich mit der Oracle Datenbank verbindet nutzen zu können müssen Sie den JDBC Treiber ojdbc6 für Oracle 11g R2 herunterladen[8] und in den Ordner ../Saiku-2.4-hetero-homogen/saiku-server/tomcat/lib kopieren und die Datei context.xml im Ordner ../Saiku-2.4-hetero-homogen/saiku-server/tomcat/webapps/HHDWService/META-INF ihrer Oracle Installation entsprechend mit Serveradresse, Benutzer und Passwort anpassen.

Die Mondrian-Schemadatei ist unter Saiku-2.4-hetero-homogen/saiku-server/tomcat/webapps/saiku/WEB-INF/classes/foodmart_test_hh zu finden.

[7] https://dev.mysql.com/downloads/connector/j/
[8] http://www.oracle.com/technetwork/apps-tech/jdbc-112010-090769.html

Drehbuch – Anweisungen zu OLAP mit hetero-homogenen DW

Schritt	Aktion, Erklärung für Studienteilnehmer
1)	**Erklärung für Studienteilnehmer:** „Online Analytical Processing (kurz: OLAP) bezeichnet die Verwendung eines Data-Warehouse, d.h., einer speziellen Datenbank für Analysezwecke, um bestimmte Fragestellungen zu beantworten. Ein Data-Warehouse organisiert Daten in sogenannten Cubes, mehrdimensionale Räume die die Betrachtung der Daten auf unterschiedlichen Abstraktionsstufen ermöglichen. Ein OLAP-Frontend erleichtert die interaktive Abfrage von Daten aus solchen Cubes“. „Sie erhalten nun eine kurze Einführung in die Arbeit mit einem OLAP-Frontend. Im Zuge dieser Einführung bekommen Sie Anweisungen, die Sie Schritt für Schritt befolgen sollen. Im Anschluss an diese Einführung sollen Sie einige von uns vorgegebene Aufgabenstellungen bewältigen um danach einen Fragebogen zur Komplexität der Navigation innerhalb des Cubes zu beantworten“.
2)	**Aktion:** Login **Erklärung für Studienteilnehmer:** „Auf dem Angabezettel finden Sie Serveradresse und Zugangsdaten für eine Webanwendung, die Ihnen ein OLAP-Frontend für die Datenanalyse zur Verfügung stellt. Rufen Sie die angegebene Serveradresse mit Ihrem Browser auf und loggen Sie sich mit den angegebenen Zugangsdaten ein“.
3)	**Aktion:** - Auswahl des Cubes „Sales_1997“ - Aufklappen aller Dimensionen **Erklärung für Studienteilnehmer:** „Wählen Sie in der Dropdown-Liste links unter ‚Cubes‘ die Option ‚Sales_1997‘ aus. Daraufhin erscheinen links unter ‚Dimensionen‘ und ‚Kennzahlen‘ mehrere Auswahlmöglichkeiten in einer Baumansicht. Erweitern Sie nun durch Klick auf die einzelnen Auswahlmöglichkeiten diese Ansicht“. „Der ausgewählte Cube enthält Verkaufsdaten. Die erfassten Kennzahlen, die diese Verkaufsdaten charakterisieren, sind ‚Profit‘, ‚Promotion Sales‘, ‚Store Cost‘, ‚Store Sales‘ und ‚Unit Sales‘“. „Die Dimensionen bestimmen den örtlichen, zeitlichen und sachlichen Gültigkeitsbereich der Kennzahlen. Dimensionen bestehen aus mehreren Levels, die hierarchisch geordnet sind. Zum Beispiel enthält die Dimension ‚Product‘ die Levels ‚Product Department Type‘, ‚Product Department‘ und ‚Product Name‘, in dieser Reihenfolge. Die hierarchische Ordnung ist so zu interpretieren, dass jeder ‚Product Name‘ zu genau einem ‚Product Department‘ gehört, jedes ‚Product Department‘ zu genau einem ‚Product Department Type‘“. „Manche Dimensionen haben alternative Level-Hierarchien. Zum Beispiel hat die Dimension ‚Product‘ eine Hierarchie ‚Product by Department‘ und eine Hierarchie ‚Brand‘“. „Die hierarchische Ordnung der Levels erlaubt die Aggregation von Kennzahlen nach den verschiedenen Levels“.
4)	**Aktion:** - Ziehen des Levels „Product Department Type“ aus der Hierarchie „Product by Department“ auf das Feld „Spalten“ - Ziehen des Levels „Store Country“ aus der Hierarchie „Store“ auf das Feld „Zeilen“ - Ziehen der Kennzahl „Profit“ auf das Feld „Spalten‘“ - Klick auf „Abfrage starten“ (grüner Pfeil) - Sortieren nach dem Level „Store Country“ - Klick auf „Abfrage starten“ - Klick auf „Achsen vertauschen“ (rechts neben „Abfrage starten“) - Klick auf „Abfrage starten“ **Erklärung für Studienteilnehmer:** „Durch Klicken und gedrückt Halten der linken Maustaste auf einen gewünschten Level in einer Dimension kann dieser Level auf das mit dem Ausdruck ‚Zeilen‘ oder ‚Spalten‘ beschriftete Feld gezogen werden. Ziehen Sie jetzt den Level ‚Product Department Type‘ auf das Feld ‚Spalten‘. Ziehen Sie danach den

	Level ‚Store Country' auf das Feld ‚Zeilen'. Auf dieselbe Weise können auch Kennzahlen ausgewählt werden. Ziehen Sie jetzt die Kennzahl ‚Profit' auf das Feld ‚Spalten'. „Wenn sie damit fertig sind klicken Sie auf den grünen Pfeil oberhalb der Zeilen- und Spaltenfelder um eine Abfrage zu starten. Daraufhin erscheint unterhalb der Zeilen- und Spaltenfelder eine Tabelle. Die Achsen dieser Tabelle sind mit Werten aus den zuvor ausgewählten Levels beschriftet. Die fettgedruckten Zellen enthalten Zeilen- und Spaltenbeschriftungen. Die übrigen Zellen enthalten aggregierte Kennzahlenwerte. Beispielsweise beträgt der gesamte Profit, den alle Multicategory-Abteilungen in allen Stores in den USA erzielt haben, 122.990,37 US-Dollar". „Klicken Sie nun zweimal auf das schwarze Doppelpfeilsymbol, das Sie auf dem blauen Kästchen mit der Beschriftung ‚Store Country' im Zeilenfeld sehen, um die Zeilen absteigend nach dem ‚Store Country' zu sortieren. Klicken Sie anschließend erneut auf den grünen Pfeil um das Ergebnis zu aktualisieren". „Klicken Sie nun auf den Button neben dem grünen Pfeil um die Achsen zu vertauschen und anschließend auf den grünen Pfeil um das Ergebnis zu aktualisieren".
5)	**Aktion:** - Setzen eines Filters auf „USA" für Level „Store Country" - Klick auf „Abfrage starten" **Erklärung für Studienteilnehmer:** „Sehen Sie sich die Levels der Dimension ‚Store' an. Für diese Dimension sind die Levels ‚Store Country', ‚Store City' und ‚Store Name' verfügbar". „Klicken Sie nun auf das Lupensymbol, das Sie auf dem blauen Kästchen mit der Beschriftung ‚Store Country' im Spaltenfeld sehen. Wählen Sie in der linken Spalte den Wert ‚USA' aus und klicken Sie auf den Button mit der einfachen spitzen Klammer, die nach rechts zeigt. Klicken Sie danach auf den OK-Button". „Sehen Sie sich nun erneut die Levels der Dimension ‚Store' an. Sie sollten jetzt einige neue Levels sehen, die ausschließlich für Stores in den USA zur Verfügung stehen. Diese Levels sind ‚Store State A' und ‚Store County A'". „Klicken Sie jetzt auf den grünen Pfeil um das Ergebnis zu aktualisieren. Nun sehen Sie nur noch die Ergebnisse für die Stores der USA".
6)	**Aktion:** - Setzen eines Filters auf „Multicategory" für Level „Product Department Type" - Ziehen des Levels „Product has Subcategory or no Subcategory" aus der Hierarchie „Product by Department" auf das Feld „Zeilen" neben „Product Department Type" - Klick auf „Abfrage starten" - Ziehen des Levels „Customer Country" auf das Feld „Zeilen" - Setzen eines Filters auf „USA" für Level „Customer Country" - Entfernen des Levels 'Product has Subcategory or no Subcategory' **Erklärung für Studienteilnehmer:** „Sehen Sie sich jetzt in der Dimension ‚Product' die Levels der Hierarchie ‚Producty by Department' an. Für diese Hierarchie sind die Levels ‚Product Department Type', ‚Product Department' und ‚Product Name' verfügbar. Außerdem ist in der Dimension ‚Product' noch eine Hierarche ‚Brand' enthalten". „Setzen Sie jetzt im Level ‚Product Department Type' den Filter auf ‚Multicategory'. Wie Sie sehen können stehen Ihnen jetzt in der Dimension ‚Product' auch die Levels ‚Product has Subcategory or no Subcategory' und ‚Product Category' zur Verfügung. Die Hierarchie ‚Brand' ist allerdings nicht mehr sichtbar, da für die zuvor getroffene Auswahl diese Hierarchie nicht anwendbar ist. „Ziehen Sie jetzt den Level ‚Product has Subcategory or no Subcategory' neben den Level ‚Product Department Type'. Beachten Sie, dass ausgewählte Levels derselben Dimension sich immer im selben Feld befinden müssen. Klicken Sie dann auf den grünen Pfeil um das Ergebnis zu aktualisieren". „In der Tabelle unterhalb der Zeilen- und Spaltenfelder sehen Sie nun den Profit, der in den USA erzielt wurde von Produkten aus Multicategory-Abteilungen aufgeschlüsselt nach Produkten die zu einer Unterkategorie gehören oder nicht". „Ziehen Sie jetzt den Level ‚Customer Country' vor den Level ‚Product Department Type' und setzen Sie bei diesem Level den Filter auf ‚USA'. Wie Sie sehen stehen Ihnen jetzt

	für die Hierarchie ‚Customers' neue Levels und die Kennzahl ‚Purchaser Count' zur Verfügung." „Sollten Sie beim Durcharbeiten der Aufgabenstellungen versehentlich einen Level auswählen den sie nicht benötigen so können Sie diesen per Drag & Drop durch Ziehen auf die Baumansicht mit den Hierarchien wieder entfernen. Diese Funktion werden Sie im Regelfall nicht benötigen."
7)	**Aktion:** - Klick auf „Task abgeschlossen" **Erklärung für Studienteilnehmer:** „Wir sind nun am Ende unserer Einführung angelangt. Klicken Sie nun auf den Button mit der Beschriftung ‚Task abgeschlossen'. Klicken Sie diesen Button später bei den Aufgabenstellungen immer nachdem Sie einen Task, der aus mehreren Unteraufgaben besteht, abgeschlossen haben".
8)	**Erklärung für Studienteilnehmer:** „Folgen Sie ab jetzt den Aufgabenstellungen, im Folgenden auch Tasks genannt, die Sie auf Ihrem Angabezettel finden. Tragen Sie Ihre Antworten in die dafür vorgesehen Felder auf dem Angabezettel ein. Bewerten Sie außerdem, nachdem Sie einen Task abgeschlossen haben, wie schwierig Sie diesen gefunden haben, auf der jeweils am Ende des Tasks angegebenen Bewertungsskala. Wenn Sie mit allen Tasks fertig sind dann füllen Sie bitte den angehängten Fragebogen aus. Beantworten Sie dabei die Fragen aus der Sicht eines Analysten der mit ähnlichen Anwendungen arbeiten muss". „Die Studienleiter beantworten während der Ausarbeitung keine Fragen. Lesen Sie den Angabezettel genau durch. Der Angabezettel ist die einzige Informationsquelle die Sie zur Lösung der Aufgaben benötigen. Die Aufgaben sind in Einzelarbeit zu lösen".

Drehbuch – Anweisungen zu OLAP mit heterogenen DW

Schritt	Aktion, Kommentar
1)	**Erklärung für Studienteilnehmer:** „Online Analytical Processing (kurz: OLAP) bezeichnet die Verwendung eines Data-Warehouse, d.h., einer speziellen Datenbank für Analysezwecke, um bestimmte Fragestellungen zu beantworten. Ein Data-Warehouse organisiert Daten in sogenannten Cubes, mehrdimensionale Räume die die Betrachtung der Daten auf unterschiedlichen Abstraktionsstufen ermöglichen. Ein OLAP-Frontend erleichtert die interaktive Abfrage von Daten aus solchen Cubes“. „Sie erhalten nun eine kurze Einführung in die Arbeit mit einem OLAP-Frontend. Im Zuge dieser Einführung bekommen Sie Anweisungen, die Sie Schritt für Schritt befolgen sollen. Im Anschluss an diese Einführung sollen Sie einige von uns vorgegebene Aufgabenstellungen bewältigen um danach einen Fragebogen zur Komplexität der Navigation innerhalb des Cubes zu beantworten“.
2)	**Aktion:** Login **Erklärung für Studienteilnehmer:** „Auf dem Angabezettel finden Sie Serveradresse und Zugangsdaten für eine Webanwendung, die Ihnen ein OLAP-Frontend für die Datenanalyse zur Verfügung stellt. Rufen Sie die angegebene Serveradresse mit Ihrem Browser auf und loggen Sie sich mit den angegebenen Zugangsdaten ein“.
3)	**Aktion:** - Auswahl des Cubes „Sales_1997“ - Aufklappen aller Dimensionen **Erklärung für Studienteilnehmer:** „Wählen Sie in der Dropdown-Liste links unter ‚Cubes‘ die Option ‚Sales_1997‘ aus. Daraufhin erscheinen links unter ‚Dimensionen‘ und ‚Kennzahlen‘ mehrere Auswahlmöglichkeiten in einer Baumansicht. Erweitern Sie nun durch Klick auf die einzelnen Auswahlmöglichkeiten diese Ansicht“. „Der ausgewählte Cube enthält Verkaufsdaten. Die erfassten Kennzahlen, die diese Verkaufsdaten charakterisieren, sind ‚Catch Costs‘, ‚Delivery Charge‘, ‚Dlv Charge per Unit‘, , Profit‘, ‚Profit – Catch Cost Ratio‘, ‚Promotion Sales‘, ‚Purchaser Count‘, ‚Store Cost‘, ‚Store Sales‘, ‚Store Sales – Catch Cost Ratio‘ und ‚Unit Sales‘“. „Die Dimensionen bestimmen den örtlichen, zeitlichen und sachlichen Gültigkeitsbereich der Kennzahlen. Dimensionen bestehen aus mehreren Levels, die hierarchisch geordnet sind. Zum Beispiel enthält die Dimension ‚Product‘ die Levels ‚Product Department Type‘, ‚Product has Subcategory or no Subcategory‘, ‚Product Department‘ usw., auf unterster Ebene befindet sich schließlich der Level ‚Product Name‘. Die hierarchische Ordnung ist so zu interpretieren, dass z.B. manche ‚Product Names‘ zu genau einem ‚Product SVUSFFF-V‘ gehören, manche ‚Product Names‘ allerdings zu genau einem ‚Product SVUSFFF-C‘. Ein weiteres Beispiel für die Interpretation der hierarchischen Ordnung der Levels dieser Dimension ist die Zuordnung von manchen ‚Product Rearing Conditions‘ zu genau einem ‚Product Department Type‘. Ob eine solche Beziehung besteht hängt vom Kontext ab“. „Manche Dimensionen haben alternative Level-Hierarchien. Zum Beispiel hat die Dimension ‚Product‘ eine Hierarchie ‚Product by Department‘ und eine Hierarchie ‚Brand‘“. „Die hierarchische Ordnung der Levels erlaubt die Aggregation von Kennzahlen nach den verschiedenen Levels“.
4)	**Aktion:** - Ziehen des Levels „Product Department Type“ aus der Hierarchie „Product by Department“ auf das Feld „Spalten“ - Ziehen des Levels „Store Country“ aus der Hierarchie „Store“ auf das Feld „Zeilen“ - Ziehen der Kennzahl „Profit“ auf das Feld „Spalten‘“ - Klick auf „Abfrage starten“ (grüner Pfeil) - Sortieren nach dem Level „Store Country“ - Klick auf „Abfrage starten“ - Klick auf „Achsen vertauschen“ (rechts neben „Abfrage starten“) - Klick auf „Abfrage starten“ **Erklärung für Studienteilnehmer:** „Durch Klicken und gedrückt Halten der linken Maus-

	taste auf einen gewünschten Level in einer Dimension kann dieser Level auf das mit dem Ausdruck ‚Zeilen' oder ‚Spalten' beschriftete Feld gezogen werden. Ziehen Sie jetzt den Level ‚Product Department Type' auf das Feld ‚Spalten'. Ziehen Sie danach den Level ‚Store Country' auf das Feld ‚Zeilen'. Auf dieselbe Weise können auch Kennzahlen ausgewählt werden. Ziehen Sie jetzt die Kennzahl ‚Profit' auf das Feld ‚Spalten'". „Wenn sie damit fertig sind klicken Sie auf den grünen Pfeil oberhalb der Zeilen- und Spaltenfelder um eine Abfrage zu starten. Daraufhin erscheint unterhalb der Zeilen- und Spaltenfelder eine Tabelle. Die Achsen dieser Tabelle sind mit Werten aus den zuvor ausgewählten Levels beschriftet. Die fettgedruckten Zellen enthalten Zeilen- und Spaltenbeschriftungen. Die übrigen Zellen enthalten aggregierte Kennzahlenwerte. Beispielsweise beträgt der gesamte Profit, den alle Multicategory-Abteilungen in allen Stores in den USA erzielt haben, 122.990,37 US-Dollar". „Klicken Sie nun zweimal auf das schwarze Doppelpfeilsymbol, das Sie auf dem blauen Kästchen mit der Beschriftung ‚Store Country' im Zeilenfeld sehen, um die Zeilen absteigend nach dem ‚Store Country' zu sortieren. Klicken Sie anschließend erneut auf den grünen Pfeil um das Ergebnis zu aktualisieren". „Klicken Sie nun auf den Button neben dem grünen Pfeil um die Achsen zu vertauschen und anschließend auf den grünen Pfeil um das Ergebnis zu aktualisieren".
5)	**Aktion:** - Setzen eines Filters auf „USA" für Level „Store Country" - Klick auf „Abfrage starten" **Erklärung für Studienteilnehmer:** „Klicken Sie nun auf das Lupensymbol, das Sie auf dem blauen Kästchen mit der Beschriftung ‚Store Country' im Spaltenfeld sehen. Wählen Sie in der linken Spalte den Wert ‚USA' aus und klicken Sie auf den Button mit der einfachen spitzen Klammer die nach rechts zeigt. Klicken Sie danach auf den OK-Button". „Klicken Sie jetzt auf den grünen Pfeil um das Ergebnis zu aktualisieren. Nun sehen Sie nur noch die Ergebnisse für die Stores der USA".
6)	**Aktion:** - Ziehen des Levels „Product Subcategory" auf „Zeilen" neben „Product Department Type" - Klick auf „Abfrage starten" - Setzen eines Filters auf „Multicategory" für Level „Product Department Type" - Ziehen des Levels „Product has Subcategory or no Subcategory" aus der Hierarchie „Product by Department" auf das Feld „Zeilen" neben „Product Department Type" - Klick auf „Abfrage starten" - Entfernen des Levels „Product Subcategory" - Klick auf „Abfrage starten" - Ziehen des Levels „Customer Country" auf das Feld „Zeilen" - Setzen eines Filters auf „USA" für Level „Customer Country" **Erklärung für Studienteilnehmer:** „Ziehen Sie jetzt den Level ‚Product Subcategory' auf das Zeilenfeld, neben den Level ‚Product Department Type'. Beachten Sie, dass ausgewählte Levels derselben Dimension sich immer im selben Feld befinden müssen. Starten Sie jetzt die Abfrage erneut. Wie Sie sehen kann es hier teilweise Null-Werte geben, da manche Produkte keine Subkategorien haben". „Setzen Sie jetzt im Level ‚Product Department Type' den Filter auf ‚Multicategory'". „Ziehen Sie jetzt den Level ‚Product has Subcategory or no Subcategory' neben den Level ‚Product Department Type'. Klicken Sie dann auf den grünen Pfeil um das Ergebnis zu aktualisieren". „Sollten Sie beim Durcharbeiten der Aufgabenstellungen versehentlich einen Level auswählen den Sie nicht benötigen so können Sie diesen per Drag & Drop durch Ziehen auf die Baumansicht mit den Hierarchien wieder entfernen. Diese Funktion werden Sie im Regelfall nicht benötigen. Entfernen Sie jetzt den Level ‚Product Subcategory' und klicken Sie erneut auf den grünen Pfeil um das Ergebnis zu aktualisieren". „In der Tabelle unterhalb der Zeilen- und Spaltenfelder sehen Sie nun den Profit, der in den USA erzielt wurde von Produkten aus Multicategory-Abteilungen aufgeschlüsselt nach Pro-

	dukten die zu einer Unterkategorie gehören oder nicht“. „Ziehen Sie jetzt den Level ‚Customer Country‘ vor den Level ‚Product Department Type‘ und setzen Sie bei diesem Level den Filter auf ‚USA‘“.
7)	**Aktion:** - Klick auf „Task abgeschlossen“ **Erklärung für Studienteilnehmer:** „Wir sind nun am Ende unserer Einführung angelangt. Klicken Sie nun auf den Button mit der Beschriftung ‚Task abgeschlossen‘. Klicken Sie diesen Button später bei den Aufgabenstellungen immer nachdem Sie einen Task, der aus mehreren Unteraufgaben besteht, abgeschlossen haben“.
8)	**Erklärung für Studienteilnehmer:** „Folgen Sie ab jetzt den Aufgabenstellungen, im Folgenden auch Tasks genannt, die Sie auf Ihrem Angabezettel finden. Tragen Sie Ihre Antworten in die dafür vorgesehen Felder auf dem Angabezettel ein. Bewerten Sie außerdem, nachdem Sie einen Task abgeschlossen haben, wie schwierig Sie diesen gefunden haben, auf der jeweils am Ende des Tasks angegebenen Bewertungsskala. Wenn Sie mit allen Tasks fertig sind dann füllen Sie bitte den angehängten Fragebogen aus. Beantworten Sie dabei die Fragen aus der Sicht eines Analysten der mit ähnlichen Anwendungen arbeiten muss“. „Die Studienleiter beantworten während der Ausarbeitung keine Fragen. Lesen Sie den Angabezettel genau durch. Der Angabezettel ist die einzige Informationsquelle die Sie zur Lösung der Aufgaben benötigen. Die Aufgaben sind in Einzelarbeit zu lösen“.

Aufgabenstellung und Fragebogen

Geben Sie für die Studie bitte folgende demografischen Daten an:

Geschlecht: m ☐ w☐

Alter: _____

Höchste abgeschlossene Ausbildung:

- ☐ Pflichtschule
- ☐ Lehre
- ☐ HAK
- ☐ HTL
- ☐ AHS
- ☐ Hochschulverwandte Schule
- ☐ Universität / Fachhochschule Bachelor
- ☐ Universität / Fachhochschule Master / Magister oder höher

Studienkennzahl des aktuellen Hauptstudiums: _____

Aufgabenstellung

Gehen Sie auf die Webseite xxx und loggen Sie sich mit den untenstehenden Zugangsdaten ein. Die Arbeitsschritte, die Sie zur Lösung der Aufgabe benötigen, werden aufgezeichnet.

Benutzername: xxx
Passwort: xxx

Gehen Sie bei der Suche nach Daten in den Tasks 1-5 so vor, dass Sie zunächst auf dem höchsten Level einer Hierarchie mit der Filterung von Daten beginnen um die Datenmenge auf relevante Informationen einzuschränken bevor Sie die Berechnung der Aggregation durchführen lassen. Beachten Sie, dass je Level immer nur ein Filter ausgewählt werden darf. Führen Sie dieses Vorgehen so lange auf den für die Suche relevanten niedrigeren Levels fort bis Sie die geforderten Daten gefunden haben.

Die einzelnen Arbeitsschritte innerhalb eines Tasks sind aufeinander aufbauend. Entfernen Sie deshalb die getroffene Auswahl an Levels, Kennzahlen und Filter während der Bearbeitung eines Tasks nur im Fehlerfall.

Beachten Sie bitte, dass in den ersten drei Tasks die Levels, Filter und Kennzahlen, welche zur Lösung benötigt werden, genau bei den einzelnen Fragen in Klammern angegeben werden. Lesen Sie also die Angabe genau und konzentriert durch.

Tragen Sie die Antwort zu jeder Frage in das dafür vorgesehene Feld ein.

Bewerten Sie bitte nachdem Sie einen Task abgeschlossen haben, wie schwierig Sie diesen gefunden haben, auf der jeweils am Ende des Tasks angegebenen Bewertungsskala. Klicken Sie nach der Bewertung der Schwierigkeit eines Tasks den Button „Task abgeschlossen“ am rechten oberen Bildschirmrand. Beachten Sie, dass ein Task aus zwei bis vier Unteraufgaben besteht.

Task 1 – Profit durch Customers aus den USA in Stores der USA

Wie hoch war der gesamte Profit (Kennzahl: „Profit“) in allen Stores der USA (Level: „Store Country“, Filter: „USA“), der durch Kunden (Dimension: „Customers“) aus den USA (Level: „Customer Country“, Filter: „USA“) generiert wurde?

Profit:_____

Wie hoch war der Profit durch Kunden die aus Kalifornien (Level: „Customer State Province“, Filter: „CA“) kommen und wie viele Kunden (Kennzahl: „Purchaser Count“) generierten diesen Profit?

Profit:_____ Purchaser Count:_____

In Kalifornien haben wir auch Geschäftskunden (Level: „Business Customer“, Filter: „YES“).
Wie hoch ist der Profit durch diese?

Profit:_____

Wie hoch ist der Profit bei Kunden mit dem Kreditrating „CCC“ (Level: „Customer Credit Rating“, Filter: „CCC“)?

CCC:_____

Wie schwierig haben Sie diesen Task gefunden (1 = sehr schwierig, 7 = sehr leicht)?

1	2	3	4	5	6	7

Klicken Sie auf den Button „Task abgeschlossen“ am rechten oberen Bildschirmrand.

Task 2 – Store Sales in mexikanischen Stores

Welcher Umsatz (Kennzahl: „Store Sales“) wurde in den mexikanischen Stores (Level „Store Country“, Filter: „Mexico“) insgesamt gemacht?

Store Sales:_____

Wie teilen sich diese Umsätze auf die mexikanischen Verwaltungsgebiete auf (Level: „Store County C“) und wie heißen diese Countys in denen das Unternehmen verkauft?

1. Store County C:_____ Store Sales:_____

2. Store County C:_____ Store Sales:_____

In welcher **Stadt** („Store City“) dieser mexikanischen Verwaltungsgebiete haben wir ein hohes Gefahrenrisiko (Level: „Store Risk“, Filter: „red“) und wie hoch ist hier der Umsatz?

Stadt:_____ Store Sales:_____

Wie schwierig haben Sie diesen Task gefunden (1 = sehr schwierig, 7 = sehr leicht)?

1	2	3	4	5	6	7

Klicken Sie auf den Button „Task abgeschlossen“ am rechten oberen Bildschirmrand.

Task 3 – Profit durch „Multicategory”-Abteilungen

Wie viel Profit wurde mit Produkten erwirtschaftet die in mehrere Kategorien unterteilt sind (Level: „Product Department Type“, Filter: „Multicategory“)?

Profit:_____

Wie viel Profit wurde mit Produkten erwirtschaftet die in weitere Subkategorien unterteilt sind (Level: „Product has Subcategory or no Subcategory“, Filter: „Subcategory“)?

Profit:_____

Wie viel Profit wurde mit der Subkategorie „Fresh Fruit“ erwirtschaftet (Level: „Product Subcategory“, Filter: „Fresh Fruit“)?

Profit:_____

Wie ist der Profit von „Fresh Fruit“ auf die einzelnen Elemente der für diese Subkategorie verfügbaren Variante des „SVUSFFF“-Levels aufgeteilt?

Profit X10UO33-P:_____ Profit Z28GL76-K:_____

Wie schwierig haben Sie diesen Task gefunden (1 = sehr schwierig, 7 = sehr leicht)?

1	2	3	4	5	6	7

Klicken Sie auf den Button „Task abgeschlossen“ am rechten oberen Bildschirmrand.

Task 4 – Profit durch „Branded" Produkte in kanadischen Stores

Wie viel Profit wurde durch Produkte mit Marke in Stores aus Kanada gemacht?

Profit:_____

Wie viel Profit wurde durch Produkte mit Marke in den einzelnen kanadischen Bundesstaaten gemacht und wie heißen diese Bundesstaaten?

1. Store State:_____ Profit:_____

2. Store State:_____ Profit:_____

Wie schwierig haben Sie diesen Task gefunden (1 = sehr schwierig, 7 = sehr leicht)?

1	2	3	4	5	6	7

Klicken Sie auf den Button „Task abgeschlossen" am rechten oberen Bildschirmrand.

Task 5 – Profitabelstes County in Stores der USA

In welchem County der US-amerikanischen Stores wurde der höchste Profit erzielt und wie hoch war dieser?

County:_____ Profit:_____

Wie viele Kunden, die aus den USA kommen, haben in diesem Store County eingekauft und wie viele Einheiten wurden verkauft?

Purchaser Count:_____ Unit Sales:_____

Bezogen auf Produkte der Multicategory-Abteilungen, und von diesen Produkten nur jene, die Subkategorien haben, welche Produkt-Subkategorie wurde im „Los Angeles County" am häufigsten verkauft? Wie viele Kunden die aus den USA kommen haben diese gekauft und wie hoch war der Profit?

Subkategorie:_____ Unit Sales:_____ Purchaser Count:_____ Profit:_____

Wie schwierig haben Sie diesen Task gefunden (1 = sehr schwierig, 7 = sehr leicht)?

1	2	3	4	5	6	7

Klicken Sie auf den Button „Task abgeschlossen" am rechten oberen Bildschirmrand.

Bewertung der Anwendung

	Stimme gar nicht zu				Stimme voll zu
1. Ich denke, dass ich dieses System gerne regelmäßig nutzen würde.	1	2	3	4	5
2. Ich fand das System unnötig komplex.	1	2	3	4	5
3. Ich denke, das System war leicht zu benutzen.	1	2	3	4	5
4. Ich denke, ich würde die Unterstützung einer fachkundigen Person benötigen, um das System benutzen zu können.	1	2	3	4	5
5. Ich fand, die verschiedenen Funktionen des Systems waren gut integriert.	1	2	3	4	5
6. Ich halte das System für zu inkonsistent.	1	2	3	4	5
7. Ich glaube, dass die meisten Menschen sehr schnell lernen würden, mit dem System umzugehen.	1	2	3	4	5
8. Ich fand das System sehr umständlich zu benutzen.	1	2	3	4	5
9. Ich fühlte mich bei der Nutzung des Systems sehr sicher.	1	2	3	4	5
10. Ich musste viele Dinge lernen, bevor ich mit dem System arbeiten konnte.	1	2	3	4	5

11. Im gesamten würde ich die Nutzerfreundlichkeit dieser Anwendung folgendermaßen bewerten:							
	Schlechtest möglich	Sehr schlecht	Schlecht	OK	Gut	Sehr Gut	Best möglich

Hetero-homogenes Mondrian-Schema

```
<?xml version="1.0"?>
<Schema name="FoodMartTestHH">
    <Dimension name="Store">
        <Hierarchy name="Store_ALL" hasAll="true" primaryKey="store_name">
            <Table name="stores_view_mhv"/>
            <Level name="Store Country" column="store_country"/>
            <Level name="Store City" column="store_city"/>
            <Level name="Store Name" column="store_name"/>
        </Hierarchy>
        <Hierarchy name="Store_USA" hasAll="true" primaryKey="store_name">
            <Table name="stores_view_usa"/>
            <Level name="Store Country" column="store_country"/>
            <Level name="Store State A" column="store_state"/>
            <Level name="Store County A" column="store_county"/>
            <Level name="Store City" column="store_city"/>
            <Level name="Store Name" column="store_name"/>
        </Hierarchy>
        <Hierarchy name="Store_Mexico" hasAll="true"
                   primaryKey="store_name">
            <Table name="stores_view_mexico"/>
            <Level name="Store Country" column="store_country"/>
            <Level name="Store County C" column="store_entidad"/>
            <Level name="Store City" column="store_city"/>
            <Level name="Store Risk" column="store_risk"/>
            <Level name="Store Name" column="store_name"/>
        </Hierarchy>
        <Hierarchy name="Store_Canada" hasAll="true"
                   primaryKey="store_name">
            <Table name="stores_view_canada"/>
            <Level name="Store Country" column="store_country"/>
            <Level name="Store State B" column="store_province"/>
            <Level name="Store County B" column="store_c_county"/>
            <Level name="Store City" column="store_city"/>
            <Level name="Store Name" column="store_name"/>
        </Hierarchy>
    </Dimension>
    <Dimension name="Storetype">
        <Hierarchy hasAll="true" primaryKey="store_name">
            <Table name="stores_view_mhv"/>
            <Level name="Storetype" column="store_type"/>
        </Hierarchy>
    </Dimension>
    <Dimension name="Time" type="TimeDimension">
        <Hierarchy hasAll="true" primaryKey="time_id">
            <Table name="time_view"/>
            <Level name="Year" column="the_year" type="Numeric"
            levelType="TimeYears"/>
            <Level name="Quarter" column="quarter"
            levelType="TimeQuarters"/>
            <Level name="Month" column="the_month" type="String"
            levelType="TimeMonths"/>
        </Hierarchy>
    </Dimension>
    <Dimension name="Product">
        <Hierarchy name="Product by Department_ALL" hasAll="true"
                   primaryKey="product_name">
            <Table name="products_view_mhv"/>
            <Level name="Product Department Type" column="depttype"/>
            <Level name="Product Department" column="product_department"/>
            <Level name="Product Name" column="product_name"/>
```

```
    </Hierarchy>
    <Hierarchy name="Product by Department_Multicategory" hasAll="true"
               primaryKey="product_name">
        <Table name="products_view_multicategory"/>
        <Level name="Product Department Type" column="depttype"/>
        <Level name="Product has Subcategory or no Subcategory"
               column="subcategorized"/>
        <Level name="Product Department" column="product_department"/>
        <Level name="Product Category" column="product_category"/>
        <Level name="Product Name" column="product_name"/>
    </Hierarchy>
    <Hierarchy name="Product by Department_Subcategorized"
               hasAll="true" primaryKey="product_name">
        <Table name="products_view_subcategorized"/>
        <Level name="Product Department Type" column="depttype"/>
        <Level name="Product has Subcategory or no Subcategory"
               column="subcategorized"/>
        <Level name="Product Department" column="product_department"/>
        <Level name="Product Category" column="product_category"/>
        <Level name="Product Subcategory"
               column="product_subcategory"/>
        <Level name="Product Name" column="product_name"/>
    </Hierarchy>
    <Hierarchy name="Product by Department_FreshFruit" hasAll="true"
               primaryKey="product_name">
        <Table name="products_view_fresh_fruit"/>
        <Level name="Product Department Type" column="depttype"/>
        <Level name="Product has Subcategory or no Subcategory"
               column="subcategorized"/>
        <Level name="Product Department" column="product_department"/>
        <Level name="Product Category" column="product_category"/>
        <Level name="Product Subcategory"
               column="product_subcategory"/>
        <Level name="Product SVUSFFF-B" column="svusfffb"/>
        <Level name="Product Name" column="product_name"/>
    </Hierarchy>
    <Hierarchy name="Brand_ALL" hasAll="true"
               primaryKey="product_name">
        <Table name="products_view_mhv"/>
        <Level name="Product is Branded or not Branded"
               column="branded"/>
        <Level name="Product Family" column="product_family"/>
        <Level name="Product Name" column="product_name"/>
    </Hierarchy>
    <Hierarchy name="Branded_Product" hasAll="true"
               primaryKey="product_name">
        <Table name="products_view_branded"/>
        <Level name="Product is Branded or not Branded"
               column="branded"/>
        <Level name="Product Family" column="product_family"/>
        <Level name="Product Brand Name" column="brand_name"/>
        <Level name="Product Name" column="product_name"/>
    </Hierarchy>
</Dimension>
<Dimension name="Promotions">
    <Hierarchy name="Promotion Mediatype" hasAll="true"
               primaryKey="promotion_id">
        <Table name="promotions_view"/>
        <Level name="Mediatype" column="media_type"/>
        <Level name="Promotion ID" column="promotion_id"/>
    </Hierarchy>
    <Hierarchy name="Promotionname" hasAll="true"
               primaryKey="promotion_id">
```

```
        <Table name="promotions_view"/>
        <Level name="Promotionname" column="promotion_name"/>
        <Level name="Promotion ID" column="promotion_id"/>
    </Hierarchy>
</Dimension>
<Dimension name="Customers">
    <Hierarchy name="Customers_ALL" hasAll="true"
               allMemberName="All Customers" primaryKey="country">
        <Table name="customers_view_mhv"/>
        <Level name="Customer Country" column="country"/>
    </Hierarchy>
    <Hierarchy name="Customers_USA" hasAll="true"
               primaryKey="fullname">
        <Table name="customers_view_usa"/>
        <Level name="Customer Country" column="country"/>
        <Level name="Customer State Province" column="state_province"/>
        <Level name="Customer City" column="city"/>
        <Level name="Customer Name" column="fullname"/>
    </Hierarchy>
    <Hierarchy name="Customers_CA" hasAll="true" primaryKey="fullname">
        <Table name="customers_view_ca"/>
        <Level name="Customer Country" column="country"/>
        <Level name="Customer State Province" column="state_province"/>
        <Level name="Business Customer" column="business_customer"/>
        <Level name="Customer City" column="city"/>
        <Level name="Customer Name" column="fullname"/>
    </Hierarchy>
    <Hierarchy name="Customers_Business" hasAll="true"
               primaryKey="fullname">
        <Table name="customers_view_business"/>
        <Level name="Customer Country" column="country"/>
        <Level name="Customer State Province" column="state_province"/>
        <Level name="Business Customer" column="business_customer"/>
        <Level name="Customer Credit Rating" column="credit_rating"/>
        <Level name="Customer City" column="city"/>
        <Level name="Customer Companyname" column="companyname"/>
        <Level name="Customer Name" column="fullname"/>
    </Hierarchy>
</Dimension>
<Cube name="Sales_1997_MHV" visible="false">
    <Table name="sales_facts_1997_mhv"/>
    <DimensionUsage name="Store" source="Store" foreignKey="store"/>
    <DimensionUsage name="Storetype" source="Storetype"
                    foreignKey="store"/>
    <DimensionUsage name="Time" source="Time" foreignKey="time"/>
    <DimensionUsage name="Product" source="Product"
                    foreignKey="product"/>
    <DimensionUsage name="Promotions" source="Promotions"
                    foreignKey="promotion"/>
    <DimensionUsage name="Customers" source="Customers"
                    foreignKey="customer"/>

    <Measure name="Unit Sales_All" column="unit_sales" aggregator="sum"
             formatString="Standard"/>
    <Measure name="Store Cost_All" column="store_cost" aggregator="sum"
             formatString="$#,###.00"/>
    <Measure name="Store Sales_All" column="store_sales"
             aggregator="sum" formatString="$#,###.00"/>
    <Measure name="Promotion Sales_All" aggregator="sum"
             formatString="$#,###.00">
        <MeasureExpression>
            <SQL dialect="mysql">(case when
            `sales_facts_1997_mhv`.`promotion` = 0 then 0 else
```

```
            `sales_facts_1997_mhv`.`store_sales` end)</SQL>
        </MeasureExpression>
    </Measure>
    <CalculatedMember name="Profit_All" dimension="Measures">
        <Formula>[Measures].[Store Sales_All] –
                 [Measures].[Store Cost_All]</Formula>
    <CalculatedMemberProperty name="FORMAT_STRING" value="$#,##0.00"/>
    </CalculatedMember>
</Cube>
<Cube name="Sales_1997_USA" visible="false">
    <Table name="sales_facts_1997_usa"/>
    <DimensionUsage name="Store" source="Store" foreignKey="store"/>
    <DimensionUsage name="Storetype" source="Storetype"
                    foreignKey="store"/>
    <DimensionUsage name="Time" source="Time" foreignKey="time"/>
    <DimensionUsage name="Product" source="Product"
                    foreignKey="product"/>
    <DimensionUsage name="Promotions" source="Promotions"
                    foreignKey="promotion"/>
    <DimensionUsage name="Customers" source="Customers"
                    foreignKey="customer"/>

    <Measure name="Purchaser Count_USA" column="customer"
             aggregator="distinct-count" formatString="#,###"/>
    <Measure name="Unit Sales_USA" column="unit_sales" aggregator="sum"
             formatString="Standard"/>
    <Measure name="Store Cost_USA" column="store_cost" aggregator="sum"
             formatString="$#,###.00"/>
    <Measure name="Store Sales_USA" column="store_sales"
             aggregator="sum" formatString="$#,###.00"/>
    <Measure name="Promotion Sales_USA" aggregator="sum"
             formatString="$#,###.00">
        <MeasureExpression>
            <SQL dialect="mysql">(case when
            `sales_facts_1997_usa`.`promotion` = 0 then 0 else
            `sales_facts_1997_usa`.`store_sales` end)</SQL>
        </MeasureExpression>
    </Measure>
    <CalculatedMember name="Profit_USA" dimension="Measures">
        <Formula>[Measures].[Store Sales_USA] –
                 [Measures].[Store Cost_USA]</Formula>
    <CalculatedMemberProperty name="FORMAT_STRING" value="$#,##0.00"/>
    </CalculatedMember>
</Cube>
<VirtualCube name="Sales_1997">
    <CubeUsages>
        <CubeUsage cubeName="Sales_1997_MHV"/>
        <CubeUsage cubeName="Sales_1997_USA"/>
    </CubeUsages>
    <VirtualCubeDimension name="Store" source="Store"
                          foreignKey="store"/>
    <VirtualCubeDimension name="Storetype" source="Storetype"
                          foreignKey="store"/>
    <VirtualCubeDimension name="Time" source="Time" foreignKey="time"/>
    <VirtualCubeDimension name="Product" source="Product"
                          foreignKey="product"/>
    <VirtualCubeDimension name="Promotions" source="Promotions"
                          foreignKey="promotion"/>
    <VirtualCubeDimension name="Customers" source="Customers"
                          foreignKey="customer"/>

    <VirtualCubeMeasure cubeName="Sales_1997_MHV"
                        name="[Measures].[Unit Sales_All]"/>
```

```
        <VirtualCubeMeasure cubeName="Sales_1997_MHV"
                            name="[Measures].[Store Cost_All]"/>
        <VirtualCubeMeasure cubeName="Sales_1997_MHV"
                            name="[Measures].[Store Sales_All]"/>
        <VirtualCubeMeasure cubeName="Sales_1997_MHV"
                            name="[Measures].[Promotion Sales_All]"/>
        <VirtualCubeMeasure cubeName="Sales_1997_MHV"
                            name="[Measures].[Profit_All]"/>

        <VirtualCubeMeasure cubeName="Sales_1997_USA"
                            name="[Measures].[Purchaser Count_USA]"
                            aggregator="distinct-count"
                            formatString="#,###"/>
        <VirtualCubeMeasure cubeName="Sales_1997_USA"
                            name="[Measures].[Unit Sales_USA]"/>
        <VirtualCubeMeasure cubeName="Sales_1997_USA"
                            name="[Measures].[Store Cost_USA]"/>
        <VirtualCubeMeasure cubeName="Sales_1997_USA"
                            name="[Measures].[Store Sales_USA]"/>
        <VirtualCubeMeasure cubeName="Sales_1997_USA"
                            name="[Measures].[Promotion Sales_USA]"/>
        <VirtualCubeMeasure cubeName="Sales_1997_USA"
                            name="[Measures].[Profit_USA]"/>
    </VirtualCube>
</Schema>
```

Heterogenes Mondrian-Schema

```
<?xml version="1.0"?>
<Schema name="FoodMart">
  <Dimension name="Store">
    <Hierarchy name="Store_ALL" hasAll="true" primaryKey="store_id">
        <Table name="stores_hh"/>
        <Level name="Store Country" column="store_country"/>
        <Level name="Store State B" column="store_province"/>
        <Level name="Store State A" column="store_state"/>
        <Level name="Store County C" column="store_entidad"/>
        <Level name="Store County A" column="store_county"/>
        <Level name="Store County B" column="store_c_county"/>
        <Level name="Store City" column="store_city"/>
        <Level name="Store Risk" column="store_risk"/>
        <Level name="Store Name" column="store_name"/>
    </Hierarchy>
  </Dimension>
  <Dimension name="Storetype">
    <Hierarchy name="Storetype" hasAll="true" primaryKey="store_id">
        <Table name="stores_hh"/>
        <Level name="Storetype" column="store_type"/>
    </Hierarchy>
  </Dimension>
  <Dimension name="Time" type="TimeDimension">
    <Hierarchy hasAll="false" primaryKey="time_id">
        <Table name="time_hh"/>
        <Level name="Year" column="the_year" type="Numeric"
               uniqueMembers="true" levelType="TimeYears"/>
        <Level name="Quarter" column="quarter" uniqueMembers="false"
               levelType="TimeQuarters"/>
        <Level name="Month" column="the_month" uniqueMembers="false"
               type="String" levelType="TimeMonths"/>
    </Hierarchy>
  </Dimension>
  <Dimension name="Product">
    <Hierarchy name="Product by Department" hasAll="true"
               primaryKey="product_id">
        <Table name="products_hh"/>
        <Level name="Product Department Type" column="depttype"/>
        <Level name="Product has Subcategory or no Subcategory"
               column="subcategorized"/>
        <Level name="Product Department" column="product_department"/>
        <Level name="Product Growing Area" column="growing_area"/>
        <Level name="Product Rearing Condition"
               column="rearing_condition"/>
        <Level name="Product Category" column="product_category"/>
        <Level name="Product Catch Area" column="catch_area"/>
        <Level name="Product Subcategory" column="product_subcategory"/>
        <Level name="Product SVUSFFF-V" column="svusfffv"/>
        <Level name="Product SVUSFFF-B" column="svusfffb"/>
        <Level name="Product SVUSFFF-C" column="svusfffc"/>
        <Level name="Product Name" column="product_name"/>
    </Hierarchy>
    <Hierarchy name="Brand" hasAll="true" primaryKey="product_id">
        <Table name="products_hh"/>
        <Level name="Product is Branded or not Branded" column="branded"/>
        <Level name="Product Family" column="product_family"/>
        <Level name="Product Brand Name" column="brand_name"/>
        <Level name="Product Name" column="product_name"/>
    </Hierarchy>
  </Dimension>
```

```
  <Dimension name="Promotions">
    <Hierarchy name="Promotion Mediatype" hasAll="true"
               primaryKey="promotion_id">
        <Table name="promotions_hh"/>
        <Level name="Media Type" column="media_type"/>
        <Level name="Promotion ID" column="promotion_id"/>
    </Hierarchy>
    <Hierarchy name="Promotionname" hasAll="true"
               primaryKey="promotion_id">
        <Table name="promotions_hh"/>
        <Level name="Promotionname" column="promotion_name"/>
        <Level name="Promotion ID" column="promotion_id"/>
    </Hierarchy>
  </Dimension>
  <Dimension name="Promotions">
  </Dimension>
  <Dimension name="Customers">
    <Hierarchy name="Customers" hasAll="true" allMemberName="All Customers"
               primaryKey="customer_id">
        <Table name="customers_hh"/>
        <Level name="Customer Country" column="country"/>
        <Level name="Customer State Province" column="state_province"/>
        <Level name="Business Customer" column="business_customer"/>
        <Level name="Customer Credit Rating" column="credit_rating"/>
        <Level name="Customer City" column="city"/>
        <Level name="Customer Companyname" column="companyname"/>
        <Level name="Customer Name" column="fullname"/>
    </Hierarchy>
  </Dimension>
<Cube name="Sales_1997">
  <Table name="sales_facts_1997_hh"/>

  <DimensionUsage name="Store" source="Store" foreignKey="store_id"/>
  <DimensionUsage name="Storetype" source="Storetype"
                  foreignKey="store_id"/>
  <DimensionUsage name="Time" source="Time" foreignKey="time_id"/>
  <DimensionUsage name="Product" source="Product" foreignKey="product_id"/>
  <DimensionUsage name="Promotions" source="Promotions"
                  foreignKey="promotion_id"/>
  <DimensionUsage name="Customers" source="Customers"
                  foreignKey="customer_id"/>

  <Measure name="Delivery Charge" column="delivery_charge" aggregator="sum"
           formatString="$#,###.00"/>
  <Measure name="Purchaser Count" column="customer_id"
           aggregator="distinct-count" formatString="#,###"/>
  <Measure name="Unit Sales" column="unit_sales" aggregator="sum"
           formatString="Standard"/>
  <Measure name="Store Cost" column="store_cost" aggregator="sum"
           formatString="$#,###.00"/>
  <Measure name="Store Sales" column="store_sales" aggregator="sum"
           formatString="$#,###.00"/>
  <Measure name="Promotion Sales" aggregator="sum"
           formatString="$#,###.00">
    <MeasureExpression>
       <SQL dialect="mysql">(case when
       `sales_facts_1997_hh`.`promotion_id` = 0 then 0 else
       `sales_facts_1997_hh`.`store_sales` end)</SQL>
    </MeasureExpression>
  </Measure>
  <Measure name="Catch Costs" column="catch_costs" aggregator="sum"
           formatString="$#,###.00"/>
  <CalculatedMember name="Store Sales - Catch Cost Ratio"
```

```
                    dimension="Measures">
     <Formula>[Measures].[Store Sales]/[Measures].[Catch Costs]</Formula>
    <CalculatedMemberProperty name="FORMAT_STRING" value="#,###0.00%"/>
  </CalculatedMember>
    <CalculatedMember name="Profit - Catch Cost Ratio"
                    dimension="Measures">
    <Formula>[Measures].[Profit] / [Measures].[Catch Costs]</Formula>
    <CalculatedMemberProperty name="FORMAT_STRING" value="#,###0.00%"/>
  </CalculatedMember>
  <CalculatedMember name="Dlv Charge per Unit" dimension="Measures">
    <Formula>[Measures].[Delivery Charge] /
             [Measures].[Unit Sales]</Formula>
    <CalculatedMemberProperty name="FORMAT_STRING" value="$#,###.00"/>
  </CalculatedMember>
  <CalculatedMember name="Profit" dimension="Measures">
    <Formula>[Measures].[Store Sales] - [Measures].[Store Cost]</Formula>
    <CalculatedMemberProperty name="FORMAT_STRING" value="$#,##0.00"/>
  </CalculatedMember>
</Cube>
</Schema>
```

Literaturverzeichnis

[1] K. Farkisch, *Data-Warehouse-Systeme kompakt: Aufbau, Architektur, Grundfunktionen*. Berlin, Heidelberg: Springer-Verlag Berlin Heidelberg, 2011.

[2] B. Neumayr, M. Schrefl, and T. Bernhard, “Hetero-Homogenous Hierarchies in Data Warehouses,” in *Seventh Asia-Pacific Conference on Conceptual Modeling (APCCM)*, 2010.

[3] C. Schütz, “Extending data warehouses with hetero-homogenous dimension hierarchies and cubes: A proof-of-concept prototype in Oracle,” Diplomarbeit, Data & Knowledge Engineering, Johannes Kepler Universität, Linz, 2010.

[4] C. Schütz, B. Neumayr, and M. Schrefl, “Integration and Reuse of Heterogenous Information: Hetero-Homogeneous Data Warehouse Modeling in the Common Warehouse Metamodel,” in *Proceedings of the Eighteenth Americas Conference on Information Systems*, Seattle, 2012.

[5] L. Rensis, “A technique for the measurement of attitudes,” (en), *Archives of Psychology*, vol. 22, no. 140, p. 55, 1932.

[6] W. H. Inmon, *Building the Data Warehouse,* 3rd ed. Hoboken: John Wiley & Sons Inc, 2002.

[7] J.-N. Mazón, J. Lechtenbörger, and J. Trujillo, “A survey on summarizability issues in multidimensional modeling,” (en), *Data & Knowledge Engineering*, vol. 68, no. 12, pp. 1452–1469, 2009.

[8] H.-G. Kemper, H. Baars, and W. Mehanna, “Informationsgenerierung, -speicherung, -distribution und -zugriff,” in *Business Intelligence – Grundlagen und praktische Anwendungen*, H.-G. Kemper, H. Baars, and W. Mehanna, Eds, Wiesbaden: Vieweg+Teubner, 2010, pp. 85–161.

[9] Springer Gabler Verlag, *Online Analytical Processing (OLAP).* Available: http://wirtschaftslexikon.gabler.de/Archiv/75659/online-analytical-processing-v8.html (2015, May. 18).

[10] E. Zangerle, *Lecture Datawarehouse Systems.* Available: https://dbis-informatik.uibk.ac.at/files/ext/lehre/ss13/DataWarehouse/8_datamining.pdf (2015, May. 18).

[11] N. Pendse and R. Creeth, “The OLAP report: succeeding with on-line analytical processing,” (en), *Business Intelligence*, 1995.

[12] *Saiku 2.x Documentation - Saiku - Meteorite Wiki.* Available: http://wiki.meteorite.bi/display/SAIK/Saiku+2.x+Documentation (2015, May. 18).

[13] Thomas S. Tullis and Jacqueline N. Stetson, “A Comparison of Questionnaires for Assessing Website Usability,” in *Usability Professional Association Conference*, Minneapolis, 2004.

[14] J. Sauro and J. S. Dumas, “Comparison of three one-question, post-task usability questionnaires,” in *the SIGCHI Conference*, pp. 1599–1608.

[15] J. Brooke, “SUS - A quick and dirty usability scale,” (en), *Usability evaluation in industry*, vol. 189, no. 194, pp. 4–7, 1996.

[16] *ISO 9241-11.* Available: http://wiki.infowiss.net/ISO_9241-11 (2015, May. 18).

[17] A. Bangor, P. T. Kortum, and J. T. Miller, “An Empirical Evaluation of the System Usability Scale,” (en), *International Journal of Human-Computer Interaction*, vol. 24, no. 6, pp. 574–594, 2008.

[18] *10 Things To Know About The Single Ease Question (SEQ): MeasuringU.* Available: http://www.measuringu.com/blog/seq10.php (2015, May. 18).

[19] *Pentaho Mondrian Documentation.* Available: http://mondrian.pentaho.com/documentation/ (2015, May. 18).

[20] J. Poole, D. Chang, D. Tolbert, and D. Mellor, *Common Warehouse Metamodel Developer's Guide*. Hoboken: John Wiley & Sons Inc, 2003.

[21] *Backbone.js.* Available: http://backbonejs.org/ (2015, May. 22).

[22] *Underscore.js.* Available: http://underscorejs.org/ (2015, May. 22).

[23] *jQuery.* Available: http://jquery.com/ (2015, May. 22).

[24] *What is Piwik? - Analytics Platform - Piwik.* Available: http://piwik.org/what-is-piwik/ (2015, Apr. 20).

[25] A. Bangor, P. T. Kortum, and J. T. Miller, “Determining what individual SUS scores mean: Adding an adjective rating scale,” (en), *Journal of usability studies*, vol. 4, no. 3, pp. 114–123, http://66.39.39.113/upa_publications/jus/2009may/JUS_Bangor_May2009.pdf, 2009.

[26] K. Lohmann and J. Schäffer, *System Usability Scale (SUS) – An Improved German Translation of the Questionnaire | Minds on WordPress.com.* Available: http://minds.coremedia.com/2013/09/18/sus-scale-an-improved-german-translation-questionnaire/ (2015, Apr. 04).

[27] M. Golfarelli, D. Maio, and S. Rizzi, “The Dimensional Fact Model: A Conceptual Model for Data Warehouses,” (ca), *Int. J. Coop. Info. Syst,* vol. 07 (2-3), pp. 215–247, 1998.