Konzeption einer Anwendung zur Ad-hoc-Sternschema-Generierung

Malte Constantinescu[1], Michael Schulz[2] and Kerstin Schneider[3]

[1]valantic Business Analytics GmbH, Beim Strohhause 17, 20097 Hamburg, Deutschland
[2]NORDAKADEMIE Hochschule der Wirtschaft, Köllner Chaussee 11, 25337 Elmshorn, Deutschland
[3]Hochschule Harz, Friedrichstrasse 57-59, 38855 Wernigerode, Deutschland

Abstract

Dimensionale Datenstrukturen sind besonders für die Anfertigung von Ad-hoc-Analysen geeignet, da diese der natürlichen Sicht entsprechen, die Fachanwender auf ihren Geschäftsbereich haben. Der Aufbau dieser Strukturen ist allerdings aufwändig. Dadurch sind sie nur für stabile Datenstrukturen geeignet, die aber gerade im Kontext von Self-Service-Analysen meist nicht gegeben sind. In diesem Beitrag soll ein Konzept beschrieben werden, durch das eine flache Tabelle (semi-) automatisch in ein Sternschema transformiert werden kann. Der Ablauf kann dabei in drei Phasen unterteilt werden: In der ersten Phase erfolgt nach Identifizierung der Datentypen jeder Tabellenspalte die Transformation der flachen Tabelle in das Sternschema. Für die Identifizierung und Zusammensetzung der Dimensionstabellen wird ein Verfahren, basierend auf der Ermittlung funktionaler Abhängigkeiten, verwendet. In der zweiten Phase sind die generierten Ergebnisse manuell zu evaluieren und ggf. zu korrigieren. In der dritten Phase werden die identifizierten Dimensions- und Faktentabellen mit Werten gefüllt und als separate Dateien ausgegeben.

Keywords

Self-Service Business Intelligence, Sternschema, Ad-hoc-Modellierung

1. Einleitung

Self-Service-Business-Intelligence-Anwendungen (SSBI-Anwendungen) erlauben es Fachanwendern, eigenständig Berichte und Analysen zu erstellen, ohne selbst über ausgeprägte technische Kenntnisse zu verfügen [1]. Eine der am häufigsten verwendeten Datenquellen sind hierbei Flat Files [2]. Dies liegt darin begründet, dass flache Strukturen häufig für den Austausch und die Bereitstellung von Daten genutzt werden. Außerdem sind sie leicht und ohne den Einsatz komplexer Softwareprodukte zu erzeugen, selbst wenn die Ursprungsdaten aus verschiedenen Quellen stammen [3]. Als Grundlage für das Ad-hoc-Reporting sind flache Strukturen jedoch weniger geeignet, da sie bei einer hohen Anzahl an Attributen schnell unübersichtlich und komplex wirken können [4].

Wesentlich geeigneter für das Erstellen von Analysen sind dimensionale Datenstrukturen, die die Aufteilung zusammengehöriger Daten in Dimensionen vorsehen. Die Ausweitung des Anwendungsbereichs von Business-Intelligence-Systemen auf operative Fragestellungen in

LWDA'21: Lernen, Wissen, Daten, Analysen September 01–03, 2021, Munich, Germany
malte.constantinescu@ba.valantic.com (M. Constantinescu); michael.schulz@nordakademie.de (M. Schulz); kschneider@hs-harz.de (K. Schneider)

den letzten beiden Jahrzehnten [5] hat jedoch zu neuen Anwendergruppen mit neuen, sich häufig ändernden Anforderungen geführt [3]. Der Aufbau und die Betreuung einer großen Anzahl dimensionaler Strukturen ist durch Business-Intelligence-Experten in der Regel nicht zu leisten, wodurch diese Form der Modellierung, trotz ihrer offenkundigen Vorteile, in den letzten Jahren an Bedeutung verloren hat. Durch das, in diesem Beitrag vorgestellte Konzept, wird es Self-Service-Anwendern möglich, eigenständig dimensionale Modelle zu erstellen, ohne auf die Expertise von Fachleuten zurückgreifen zu müssen. Dadurch soll es dieser Anwendergruppe möglich werden, Strukturen in Daten schneller zu verstehen und ihre Analyseperformance zu verbessern.

In Kapitel 2 wird zunächst die Relevanz für die Erstellung von Ad-hoc-Sternschemas erläutert. Anschließend werden die Ziele der zu konzipierenden Arbeit dargestellt (Kapitel 3). Die Konzeption der Anwendung wird in Kapitel 4 vorgestellt. In den einzelnen Abschnitten dieses Kapitels werden die drei Phasen der automatischen Sternschema-Modellierung präsentiert. In Kapitel 5 erfolgt eine Zusammenfassung der Ergebnisse sowie ein Ausblick auf zukünftige Anwendungsmöglichkeiten der erarbeiteten Lösung.

2. Status Quo

Hänel und Schulz haben in einem Experiment mit einer einfachen SSBI-Anwendung einen Vorteil von dimensionalen Strukturen gegenüber flachen und transaktionsorientierten Strukturen identifiziert [3]. Auch wenn sich diese Erkenntnis in einem späteren Experiment nicht vollständig bestätigen ließ ([6]) und auch andere ähnlich aufgebaute Experimente nicht zu einheitlichen Ergebnissen führten ([7] [8] [9] [10] [11]), so scheint in der Praxis der Vorteil einer multidimensionalen Sicht auf Daten, vor allem bei der Analyse von komplexen und für die Analysierenden unbekannten Unternehmensdaten, wahrgenommen zu werden. Codd et al. [12] sowie Kimball und Ross [13] sehen den Vorteil der dimensionalen Datenstrukturen vor allem darin begründet, dass sie der natürlichen Sicht entsprechen, die die Fachanwender auf ihren Geschäftsbereich haben. Auch Anbieter von aktuellen SSBI-Anwendungen wie Microsoft Power BI oder QlikView empfehlen mit dem Sternschema die Verwendung eines dimensionalen Datenmodells [14] [15]. Die Gründe hierfür liegen in einer optimierten Performance sowie einer höheren Benutzerfreundlichkeit gegenüber anderen Strukturen [14]. Der Nachteil dimensionaler Strukturen ist jedoch, dass ihr Aufbau aufwändig ist und zugleich heutzutage häufig sehr schnell erfolgen muss, da sich der Anwendungsbereich von Business-Intelligence-Systemen in den letzten Jahrzehnten von strategischen auf operative Anwendungsbereiche mit sich schnell ändernden Problemstellungen ausgeweitet hat.

Eine mehrdimensionale Sicht auf Daten kann in einer relationalen Struktur durch ein Sternschema hergestellt werden [13]. Dieses Datenmodell setzet sich aus einer Faktentabelle und mehreren Dimensionstabellen zusammen.

Ein wesentlicher Bestandteil des SSBI-Ansatzes ist eine geeignete Anwendung, durch die Fachanwender in die Lage versetzt werden, eigenständig Berichte und Analysen zu erstellen, ohne über technische Kenntnisse zu verfügen. Diese Anwendungen erlauben den Import verschiedener Datenquellen und verfügen über die Möglichkeit der eigenständigen Modellierung von Daten. Wird als Datenquelle auf eine flache Datenstruktur (z.B. Flat File) zugegriffen, so

ist der Aufbau einer mehrdimensionalen Sicht auf die Daten in vielen Anwendungen zwar möglich, allerdings sehr zeitaufwändig und bedingt ein Verständnis über die zugrundeliegenden Strukturen, das von Fachanwendern in der Regel nicht erwartet werden kann.

3. Ziele des Verfahrens zur Sternschema-Generierung

Das Hauptziel der zu konzipierenden Anwendung ist es, eine flache Tabelle in eine dimensionale Datenmodell-Struktur (Sternschema) zu überführen. Der Anwendende soll dabei die Zusammensetzung des Sternschemas, insbesondere die der Dimensionen, ad hoc und ohne tiefgehende Kenntnisse der Datenmodellierung an seine Bedürfnisse anpassen können. Zusammenfassend soll die Anwendung über folgende Funktionalitäten verfügen:

(1) Die Anwendenden können eine flache Tabelle als Datei in die Anwendung laden. Die Tabelle wird anschließend automatisch in ein Sternschema transformiert. Hierzu sollen die Dimensionen aus der flachen Tabelle extrahiert sowie eine Faktentabelle, bestehend aus den Primärschlüsseln der Dimensionstabellen und den Kennzahlen, generiert werden.

(2) Die Fakten- und Dimensionstabellen sollen als separate Dateien wieder ausgegeben werden. Diese können dann in eine SSBI-Anwendung (z.B. Tableau, Microsoft Power BI oder QlikView) geladen werden. Alternativ oder zusätzlich können, gerade bei größeren Quelldatensets, SQL-DDL- und DML-Skripte zur Definition und Befüllung der Fakten- und Dimensionstabellen generiert werden, um das Speichern in einer Datenbank zu ermöglichen und die Abfragegeschwindigkeiten zu erhöhen.

(3) Enthält die flache Tabelle Datumswerte, soll auch eine Zeitdimension erstellt werden. Sind mehrere Datumswerte (z.B. Verkaufsdatum und Lieferdatum) in der Faktentabelle vorhanden, sollen die Anwendenden bestimmen können, auf welches Datum sich die Zeitdimension bezieht oder ob mehrere Zeitdimensionen erstellt werden sollen.

(4) Da die Sternschema-Modellierung immer auch von der zu untersuchenden Problemstellung abhängt, sollen die Anwender die Möglichkeit bekommen, die identifizierten Dimensionen noch einmal anzupassen. D.h. Dimensionen oder Dimensionsattribute, die für eine nachfolgende Analyse nicht benötigt werden, können entfernt werden. Falls Attribute den falschen Dimensionen zugeordnet wurden, soll auch dies korrigiert werden können.

4. Konzeption

4.1. Ablauf der Sternschema-Generierung

Der Ablauf der Sternschema-Generierung kann in drei Phasen aufgeteilt werden (siehe Abbildung 1). Die Sternschematransformation beginnt damit, dass die Anwendenden die flache Tabelle als Datei (z.B. CSV-Datei) in die Anwendung laden und endet damit, dass die Dimensions- und Faktentabelle als separate Dateien ausgegeben werden. Alternativ oder zusätzlich werden SQL-DDL- und DML-Skripte zur Definition und Befüllung der Dimensions- und Faktentabellen erstellt.

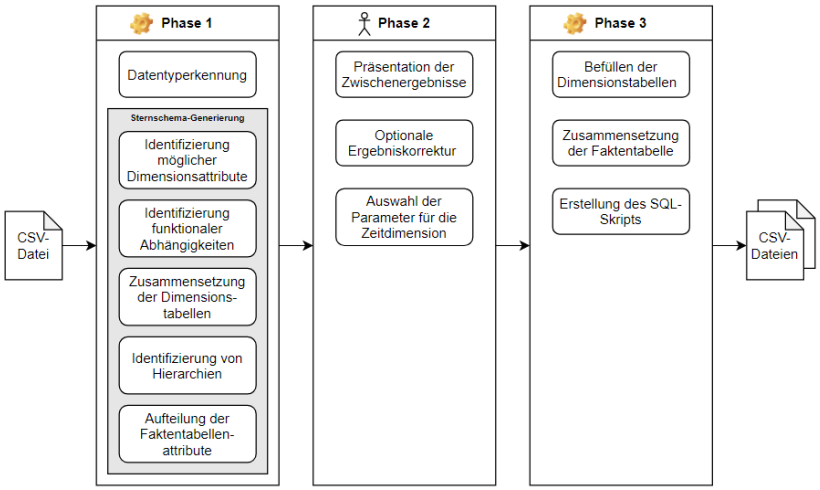

Figure 1: Ablauf der Sternschema-Modellierung

4.2. Phase 1

In der ersten Phase stehen besonders die Datentyperkennung und Sternschema-Generierung im Vordergrund:

Datentyperkennung: Eine wesentliche Voraussetzung zur Generierung des SQL-Skripts ist die Identifizierung der Datentypen einer jeden Tabellenspalte. Hierzu gilt es herauszufinden, ob in einer Tabellenspalte Zeichenketten, ganze Zahlen, Fließkommazahlen, Datumswerte oder Uhrzeiten enthalten sind.

Identifizierung möglicher Dimensionsattribute: Die Zusammensetzung der Dimensionstabellen basiert auf der Identifizierung funktionaler Abhängigkeiten. Bestehen funktionale Abhängigkeiten zwischen Attributen sollen sie zu einer Dimensionstabelle zusammengefasst werden. Bevor dies durchgeführt werden kann, müssen zwei Aspekte beachtet werden:

(1) Schlüsselkandidaten. Die flache Ausgangstabelle verfügt über einen oder mehrere Schlüsselkandidaten: Da alle Attribute von einem Schlüsselkandidaten funktional abhängig sind, würde nur eine Dimensionstabelle erkannt werden. Um dieses Szenario zu verhindern, muss die flache Tabelle auf das Vorhandensein von Schlüsselkandidaten überprüft werden. Hierzu wird für jedes Attribut die Anzahl der verschiedenen Werte ermittelt. Falls dieser Wert der Gesamtanzahl an Datensätzen entspricht, handelt es sich bei diesem Attribut um einen Schlüsselkandidaten. Dieses Attribut kann als möglicher Schlüsselkandidat einer Dimension ausgeschlossen werden. Dieses Vorgehen ist nur relevant, sofern Schlüsselkandidaten existieren, die genau ein Schlüsselattribut besitzen. Zusammengesetzte Schlüsselkandidaten verhindern nicht die Erstellung mehrerer Dimensionstabellen.

(2) Degenerierte Dimensionen. Eine degenerierte Dimension ist ein Dimensionsschlüssel ohne eigene Dimensionstabelle [13]. Ein Beispiel, welches die Problematik mit einem Teil der degenerierten Dimensionen zeigt, ist das Attribut Rechnungsnummer. Einer Rechnung kann immer genau ein Kunde zugewiesen werden, d.h. zwischen den Attributen Rechnungsnummer

und Kundennummer liegt eine funktionale Abhängigkeit vor, die die korrekte Zusammensetzung der Dimension Kunde verhindert. Um degenerierte Dimensionen zu identifizieren, muss für jedes Attribut die Anzahl der unterschiedlichen Werte ermittelt werden. Dieser Wert ist anschließend ins Verhältnis zu der Gesamtanzahl an Datensätzen zu setzen. Je größer das Verhältnis ist, desto größer ist die Wahrscheinlichkeit, dass es sich bei diesem Attribut um eine degenerierte Dimension handelt. Zur Trennung zwischen normalen Dimensionsattribut oder degenerierter Dimension wird, aufgrund von Versuchen mit verschiedenen Testdatensätzen, ein Grenzwert von 0,2 festgelegt. Dieser ist im konkreten Anwendungsfall ggf. anzupassen. Überschreitet ein Verhältnis diesen Grenzwert, so kann es sich bei dem Attribut um eine mögliche degenerierte Dimension handeln. Da dieser Grenzwert nicht allgemeingültig ist und zudem nur darauf hindeutet, dass es sich bei einem Attribut um eine degenerierte Dimension handelt, dürfen die entsprechenden Attribute nicht von vornherein als normales Dimensionsattribut ausgeschlossen werden. Stattdessen sollten sie als mögliche degenerierte Dimension gekennzeichnet werden, sodass der Anwendende selbst entscheiden kann, ob es sich bei einem Attribut um eine degenerierte Dimension oder um den Teil einer vollständigen Dimension handelt.

Identifizierung funktionaler Abhängigkeiten zwischen den Dimensionsattributen: In diesem Schritt werden alle funktionalen Abhängigkeiten, die zwischen Attributen vorliegen, ermittelt. Um die Darstellung des Konzepts auf das Wesentliche zu beschränken, wird ausschließlich die Untersuchung der funktionalen Abhängigkeiten zwischen Attributpaaren dargestellt. Zur Identifikation von Dimensionstabelle mit zusammengesetzten Schlüsselkandidaten ist das Vorgehen entsprechend zu erweitern. Zur Identifizierung aller Abhängigkeitspaare werden alle Zweierkombinationen der möglichen Dimensionsattribute auf funktionale Abhängigkeit überprüft. Wird eine funktionale Abhängigkeit zwischen zwei Attributen identifiziert, so werden diese der Menge der funktionalen Abhängigkeitspaare hinzugefügt. Das beschriebene Verfahren ist in Abbildung 2 zusammenfassend dargestellt.

```
    finde_funktionale_Abhängigkeiten(t)
1   Eingabe: Flache Tabelle t
2   Ausgabe: Liste mit funktionalen Abhängigkeitspaaren l
3   k := 0
4   while k < (Länge von t) -1 do
5       j := k+1
6       while j < Länge von t do
7           f := Liste zum Speichern der Abhängigkeitspaare
8           {Wenn funktionale Abhängigkeit vorhanden → Rückgabe der beiden Attribute}
9           f = finde_Abhängigkeit(Attribut von t an Stelle k, Attribut von t an Stelle j)
10          if ap enthält zwei Attribute then
11              Hinzufügen von f zu l
12          endif
13          j = j+ 1
14      endwhile
15      k = k+1
16  endwhile
```

Figure 2: Algorithmus zur Identifizierung der funktionalen Abhängigkeitspaare

Zusammensetzung der Dimensionstabellen: Basierend auf den ermittelten Abhängigkeitspaaren werden in diesem Schritt die Dimensionstabellen gebildet. Die Zusammensetzung basiert darauf, dass jede Dimensionstabelle über ein Primärschlüsselattribut verfügt, über das die Granularität der Dimension definiert wird. D.h. alle Attribute, die über dasselbe bestimmende

Attribut verfügen, werden zu einer möglichen Dimension zusammengefasst.

Beispiel: Im vorangegangenen Schritt wurden u.a. die folgenden funktionalen Abhängigkeitspaare identifiziert:

Kundennummer → Kundenname
Kundennummer → Kundenadresse
Kundennummer → MarktsegmentID
Kundennummer → Marktsegment

Basierend auf identifizierten funktionalen Abhängigkeiten kann die Dimension *Kunde* bestehend aus den Attributen *Kundennummer*, *Kundenname*, *Kundenadresse*, *MarktsegmentID* und *Marktsegment* identifiziert werden.

Identifizierung von Hierarchien: Da bei der Identifizierung funktionaler Abhängigkeiten zwischen den Dimensionsattributen auch transitive Abhängigkeiten ermittelt werden, können basierend auf diesen, auch Hierarchien innerhalb einer Dimension identifiziert werden. Neben der Möglichkeit, dass keine Hierarchie vorliegt, kann zwischen drei Hierarchietypen unterschieden werden:

1. **Einfache Hierarchien**: Bestehen aus einer linearen Abfolge von mindestens zwei Dimensionshierarchiestufen [16]. Die einzelnen Hierarchiestufen sind über Aggregationsbeziehungen miteinander verbunden [17].
2. **Parallele Hierarchien**: Liegen vor, wenn einer Dimension mehrere Hierarchien zugeordnet werden können, die unterschiedliche Analysekriterien berücksichtigen [17].
3. **Unbalancierte Hierarchien**: Auf Ausprägungsebene lassen sich Hierarchien als Baumstruktur darstellen. Eine unbalancierte Hierarchie liegt vor, sobald sich die Länge der Zweige mindestens um den Wert Eins unterscheidet [16].

Da unbalancierte Hierarchien nicht auf der Basis funktionaler Abhängigkeiten ermittelt werden können und in einer flachen Tabelle ohnehin nicht abbildbar sind, wird sich in diesem Schritt nur auf die Identifizierung von einfachen und parallelen Hierarchien beschränkt. Die Identifizierung dieser, wird im Folgenden anhand von zwei Beispielen erläutert:

Beispiel (einfache Hierarchie): Aus den Attributen *KundenID*, *Kundenname*, *Kundenname*, *MarktsegmentID*, *Marktsegment*, *MarktsektorID* und *Marktsektor* können die in Abbildung 3 (linke Seite) dargestellten funktionalen Abhängigkeitspaare ermittelt werden, basierend darauf kann die auf der rechten Seite der Abbildung dargestellte Hierarchie abgeleitet werden.

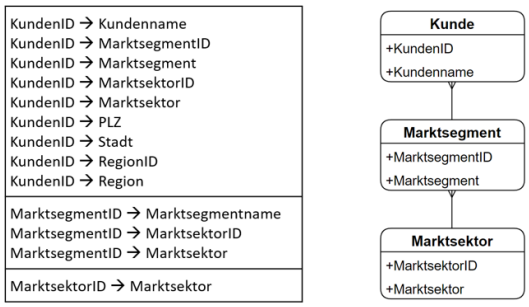

Figure 3: Beispiel für einfache Hierarchie

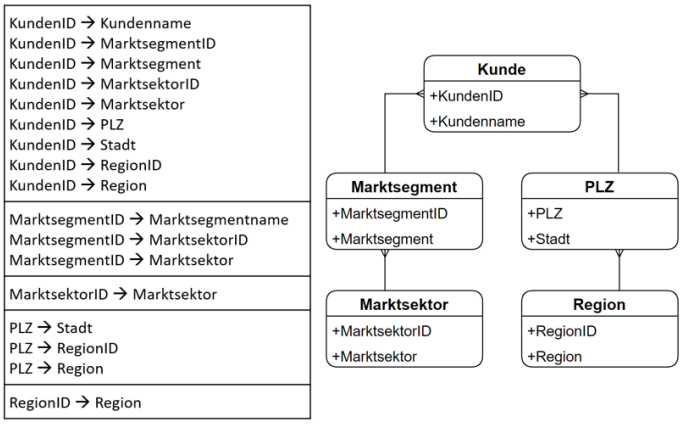

Figure 4: Beispiel für parallele Hierarchie

Beispiel (parallele Hierarchie): Kommen zu den Attributen aus dem ersten Beispiel noch die Attribute *PLZ*, *Stadt*, *RegionID* und *Region* hinzu, so können die in Abbildung 4 (linke Seite) dargestellten funktionalen Abhängigkeitspaare identifiziert werden, die abgeleitete Hierariche ist auf der rechten Seite der Abbildung dargestellt.

Aufteilung der Faktentabellenattribute: Im letzten Schritt der Sternschema-Generierung gilt es die Fakten, Datumswerte, Uhrzeiten und degenerierte Dimensionen zu identifizieren. Die Kandidatenmenge hierzu umfasst alle Attribute, die zu keinem der im Schritt *Identifizierung funktionaler Abhängigkeiten zwischen den Dimensionsattributen* ermittelten Abhängigkeitspaare gehören. Zur Trennung zwischen Fakten, Datumswerten, Uhrzeiten und degenerierte Dimensionen werden folgende Regeln definiert:

- **Datumswerte**: Attribute, die Datumswerte oder Uhrzeiten enthalten
- **Degenerierten Dimensionen**: Attribute, die Zeichenketten enthalten
- **Fakten**: Attribute, die ganze Zahlen oder Fließkommazahlen enthalten Eine Unterscheidung von Fakten und degenerierten Dimensionen, deren Attribut numerische Werte enthält, ist ohne das Domänenwissen der Anwendenden unmöglich. Falschzuordnungen müssen daher, wie im nächsten Kapitel beschrieben, manuell korrigiert werden.

4.3. Phase 2

In der zweiten Phase steht die Evaluierung und optionale Anpassung der Ergebnisse der Sternschema-Generierung durch die Anwendenden im Vordergrund:

Präsentation der Zwischenergebnisse: Die Ergebnisse der automatischen Strukturierung sind den Anwendenden in geeigneter Form zu präsentieren, ein beispielhaftes Mockup ist in Abbildung 5 dargestellt. Jede Dimension wird darin als separates Listenfeld dargestellt, in dem die jeweiligen Dimensionsattribute enthalten sind. Der Dimensionsschlüssel befindet sich dabei an erster Stelle. Die **Fakten**, **Datumswerte** und **degenerierte Dimensionen** werden in den gleichnamigen Listenfeldern präsentiert.

Optionale Ergebniskorrektur:Da eine Falschzuordnung von Attributen zu Dimensionen nicht ausgeschlossen werden kann, müssen die Anwendenden die Möglichkeit erhalten, falschen

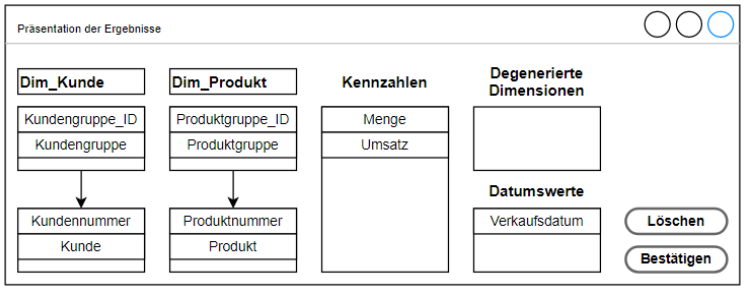

Figure 5: Benutzeroberfläche zum Anpassen der Ergebnisse

Zuordnungen aufzulösen und die Attribute den richtigen Dimensionstabellen zuzuordnen. Hierzu müssen die Attribute z. B. per Drag n Drop in das Listenfeld der richtigen Dimension verschoben werden können. Auch das Verschieben von Attributen in die Listenfelder Fakten, Datumswerte und degenerierte Dimensionen muss hierbei möglich sein. Außerdem müssen für die betrachtete Problemstellung irrelevante Attribute gelöscht werden können – im Mockup durch den Button Löschen visualisiert.

Auswahl der Parameter für die Zeitdimension: Die Struktur der Zeitdimension kann im Gegensatz zu den Strukturen der anderen Dimensionen nicht direkt aus der flachen Tabelle übernommen werden, sondern muss aus einem Datumsattribut generiert werden. Eine Zeitdimension enthält alle Datumswerte oder Uhrzeiten innerhalb eines bestimmten Zeitraums sowie eine Menge beschreibender Attribute (z.B. Monat, Quartal etc.), die aus dem gegebenen Datum abgeleitet werden können und dem Anwendenden zusätzliche Analysemöglichkeiten bieten [16]. Hierzu muss der Anwendende in diesem Schritt bestimmen können, auf welche Datums- oder Uhrzeitattribut sich die Zeitdimension bezieht und zum anderen die Granularität der Zeitdimension festlegen können.

4.4. Phase 3

In der dritten Phase erfolgt die Befüllung der Dimensions- und Faktentabellen mit Daten sowie die Generierung eines SQL-Skriptes:

Befüllen der Dimensionstabellen: Die Dimensionsattribute und die entsprechenden Werte aus der flachen Tabelle werden extrahiert und in jeweils eine Zieldatei geladen. Abhängig von der Implementierung sind hier ggf. auch Funktionen zur Datentypkonvertierung vorzusehen. Zur Befüllung der Zeitdimension wird zunächst das minimale und maximale Datum aus der entsprechenden Spalte der flachen Ausgangstabelle ermittelt. Die Granularität der Zeitdimension ergibt sich aus dem, vom Anwendenden ausgewählten Parameter (vgl. Phase 2).

Zusammensetzung der Faktentabelle: Auch die Attribute der Faktentabelle werden in einer Datei ausgegeben. Für die Zusammensetzung der Faktentabelle müssen die Primärschlüssel der Dimensionstabellen, Kennzahlen, Datumswerte und degenerierte Dimensionen aus der flachen Ausgangstabelle ausgewählt werden.

SQL-Skript-Erstellung: In diesem Schritt werden die SQL-DDL-Befehle zur Definition der Dimensions- und Faktentabellen inklusive deren Primär- und Fremdschlüsseln erstellt. Auch die

SQL-DML-Befehle zur Befüllung dieser Tabellen werden gebildet. Sämtliche SQL-Statements erfordern die Angabe von Tabellennamen. Falls den Dimensionstabellen im Schritt Optionale Ergebniskorrektur ein Name zugewiesen wurde, so wird dieser verwendet. Trifft die nicht zu, so wird der Name des Primärschlüsselattributs herangezogen. Mit Ausnahme von Besonderheiten in der Datumsdimension, basiert jedes Dimensions- und Faktenattribut auf einem Attribut der Ursprungstabelle. Die Namen können daher übernommen werden, der Datentyp wurde bereits zu Beginn des beschriebenen Prozesses (vgl. Phase 1) ermittelt.

5. Fazit und Ausblick

In diesem Beitrag wurde das Konzept einer Anwendung vorgestellt, durch die eine flache Ausgangsstruktur ad hoc und (semi-)automatisch in ein Sternschema transformiert werden kann. Durch eine erste Implementierung des Konzeptes und deren Anwendung auf verschiedene Beispieldatensets konnte die Funktionsfähigkeit des Ansatzes nachgewiesen werden. Grundvoraussetzung für eine korrekte Datenmodellierung ist jedoch eine Ausgangsstruktur mit einer ausreichenden Zahl an Datensätzen, um auszuschließen, dass zufällig vorhandene funktionale Abhängigkeiten das Ergebnis beeinträchtigen. Auch eine hohe Datenqualität ist Bedingung für eine korrekte Modellierung. Durch eine leichte Erweiterung des vorgestellten Konzeptes, könnte es allerdings auch gerade dazu genutzt werden, Gelegenheitsanwendern die Möglichkeit zu geben, Unstimmigkeiten in den Zusammenhängen der verwendeten Daten zu identifizieren.

Zukünftig soll die Integration der vorgestellten Datenmodellierungskomponente über zu entwickelnde Schnittstellen in SSBI-Software erfolgen. Durch diese Erweiterung des Konzeptes entfällt der momentan nötige Zwischenschritt, die erstellten Dateien manuell in ein Tool zu laden. Anwendende können dann ohne zusätzliche Vorarbeiten direkt mit dem Stern-Schema arbeiten. Weiterhin soll mit Hilfe von Experimenten der Nutzen des Ansatzes für Gelegenheitsanwender ohne tiefergehende Datenkompetenzen überprüft werden. Auch wenn der generelle Vorteil einer mehrdimensionalen Sicht auf die Daten in Laborexperimenten bisher noch nicht eindeutig nachgewiesen werden konnte, so scheinen doch Problemstellungen zu existieren, bei denen diese Struktur zu bevorzugen ist (vgl. z.B. [5]). Die unstrittige Bedeutung von Stern-Schemas für die Praxis wird auch dadurch hervorgehoben, dass Hersteller von SSBI-Software sie häufig zur Verwendung empfehlen [13], [14]. Eine Implementierung des vorgestellten Ansatzes böte den Anwendenden mindestens die Möglichkeit, im individuellen Analyseszenario zwischen einer flachen und einer mehrdimensionalen Struktur wählen zu können. Welche Merkmale Datenquellen aufweisen müssen, damit eine (semi-)automatische Erstellung eines Stern-Schemas gelingt und in der Anwendung einen Nutzen schafft, gilt es ebenfalls mit Hilfe von Experimenten zu untersuchen.

Zukünftige Anwendungsszenarien des Konzeptes sind nicht nur auf die, in diesem Beitrag in den Fokus gestellten Ad-hoc-Analysen beschränkt. Modellierte Stern-Schema-Strukturen können von einzelnen aber auch von mehreren Benutzern wiederverwendet werden, wenn neue Daten wiederkehrend in bekannten Strukturen verfügbar werden. Auch ist der Ansatz für Data-Warehouse-Entwickler nutzbar, um Anforderungen der Fachanwender für den Aufbau von standardisierten Stern-Schemas, ohne den Aufwand manueller Datentransformation, ad hoc zu überprüfen.

References

[1] M. Daradkeh, R. Al-Dwairi, Self-service business intelligence adoption in business enterprises: the effects of information quality, system quality, and analysis quality, in: Operations and Service Management: Concepts, Methodologies, Tools, and Applications, IGI Global, 2018, pp. 1096–1118.

[2] T. Grosser, R. Tischler, Data Preparation im Fachbereich – aus Rohdaten den Treibstoff für Ihr Unternehmen gewinnen, CXP Group, 2017.

[3] T. Hänel, M. Schulz, Is there still a need for multidimensional data models?, in: Proceedings of the 22nd European Conference on Information Systems, ECIS 2014, 2014.

[4] D. Moody, M. Kortink, From enterprise models to dimensional models: a methodology for data warehouse and data mart design, in: Proceedings of the International Workshop on Design and Management of Data Warehouses, 2000, p. 5.

[5] P. K. C. S. C. Böhringer, M.and Gluchowski, Business intelligence perspective on the future internet, in: Proceedings of the 16th Americas Conference on Information Systems, 2010, p. 267.

[6] M. Schulz, P. Alpar, P. Winter, Should data structures look flat for end users?, Information Systems Management 37 (2000) 150–169.

[7] K. Dowling, D. Schuff, R. D. St. Louis, Dimensional data models versus entity relationship models: Does it make a difference to end-users?, in: Proceedings of the 7th Americas Conference on Information Systems, 2001, p. 80.

[8] K. Corral, D. Schuff, R. D. St. Louis, The impact of alternative diagrams of the accuracy of recall: A comparison of star-schema diagrams and entity-relationship diagrams, Decision Support Systems 42 (2006) 450–468.

[9] D. Vujošević, I. Kovačević, M. Suknović, N. Lalić, A comparison of the usability of performing ad hoc querying on dimensionally modeled data versus operationally modeled data, Decision Support Systems 54 (2012) 185–197.

[10] S. M. Yusof, F. Sidi, Relational model vs. dimensional model – further experimentation on understandability of the two schemas, in: Proceedings of the 1st Malaysian National Conference on Databases, 2014, pp. 40–45.

[11] D. Schuff, K. Corral, O. Turetken, Comparing the understandability of alternative data warehouse schemas: An empirical study, Decision Support Systems 52 (2011) 9–20.

[12] E. Codd, S. Codd, C. Salley, Providing OLAP (on-line analytical processing) to user-analysts. An IT Mandate, Arbor Software Corporation, 1993.

[13] R. Kimball, M. Ross, The data warehouse toolkit: The definitive guide to dimensional modeling, John Wiley Sons, Indianapolis, 2013.

[14] Microsoft, Informationen zum sternschema und der wichtigkeit für power bi, 2019. URL: http://www.poker-edge.com/stats.php.

[15] QlikView, Sternschema, o.J. URL: https://www.qlikview-info.de/im-skript/sternschema.

[16] D. Schnider, C. Jordan, P. Welker, J. Wehner, Data Warehouse Blueprints - Business Intelligence in der Praxis, Hanser Verlag, München, 2006.