dani SCHNIDER
claus JORDAN
peter WELKER
joachim WEHNER

DATA WAREHOUSE BLUEPRINTS

Business Intelligence in der Praxis

HANSER

trivadis
makes IT easier.

Schnider / Jordan / Welker / Wehner

Data Warehouse Blueprints

Dani Schnider
Claus Jordan
Peter Welker
Joachim Wehner

Data Warehouse Blueprints

Business Intelligence in der Praxis

HANSER

Bibliografische Information der Deutschen Nationalbibliothek
Die Deutsche Nationalbibliothek verzeichnet diese Publikation in der Deutschen Nationalbibliografie; detaillierte bibliografische Daten sind im Internet über <*http://dnb.d-nb.de*> abrufbar.

Lektorat: Sylvia Hasselbach
Herstellung: Irene Weilhart
Copy editing: Sandra Gottmann, Münster-Nienberge
Umschlagdesign: Marc Müller-Bremer, *www.rebranding.de,* München
Umschlagrealisation: Stephan Rönigk
Gesamtherstellung: Kösel, Krugzell
Ausstattung patentrechtlich geschützt. Kösel FD 351, Patent-Nr. 0748702
Printed in Germany

Print-ISBN: 978-3-446-45075-2
E-Book-ISBN: 978-3-446-45111-7

Inhalt

Geleitwort

Von Dr. Carsten Bange, Gründer und Geschäftsführer des Business Application Research Centers (BARC), Teil des europäischen Analystenhauses CXP Group.

Noch ein Buch über Data Warehousing? Ist darüber in den vergangenen 25 Jahren nicht genug geschrieben worden? Ich gebe zu, ich war skeptisch als die Autoren mich baten, ein Vorwort zu verfassen. Insbesondere auch, da wir in unserer täglichen Praxis als Marktanalysten eine deutlich wachsende Kritik vieler Unternehmen an ihrem Data Warehouse wahrnehmen. Insbesondere die Anwender verlangen nach Änderungen, um ihren veränderten Anforderungen Rechnung zu tragen. Die letzte BARC-Anwenderbefragung zu diesem Thema[1] zeigt deutlich, was den Veränderungsbedarf treibt: 62 % der 323 befragten BI- und Data-Warehouse-Verantwortlichen sehen sich mit deutlich erhöhten Erwartungen in den Fachbereichen konfrontiert, 51 % verstehen dabei die schnellere Veränderung von Geschäftsprozessen als wesentlichen Treiber für Anpassungen an Datenmanagement-Konzepten und 45 % erfahren eine Unzufriedenheit mit der benötigten Zeit, um neue Anforderungen im Data Warehouse umzusetzen.

Vielen Unternehmen wird also immer klarer, dass sie die etablierten Data-Warehouse-Systeme so nicht mehr weiterbetreiben können, sondern hinsichtlich der Prozesse und der Organisation, IT-Architektur und eingesetzte Technologien und Werkzeuge komplett überdenken müssen.

Das vorliegende Buch liefert hierzu einen guten Beitrag und legt seinen Fokus dabei auf Methodik und Technologie. Es trägt eine große Menge von Erfahrungen zu „Best Practice"-Anleitungen zusammen, die helfen, das eigene Projekt auf eine solide Basis zu stellen und typische Fehler zu vermeiden. Es behandelt dabei auch neue Technologien, z. B. aus dem Hadoop- und NoSQL-Umfeld, die eine interessante Ergänzung der etablierten und ausgereiften Datenbank- und Datenintegrationstechnologien sein können. Die Autoren bieten damit Entwicklern, BI- und Data-Warehouse-Verantwortlichen ein solides methodisches Fundament, um die Informationsversorgung zur Entscheidungsfindung in Unternehmen erfolgreich aufzubauen.

Es bleibt dann im Unternehmen die wichtige Aufgabe, die verfügbaren Technologien in eine anforderungsgerechte Organisation einzubetten. Gerade Agilität und Flexibilität sind hier

[1] s. BARC-Anwenderbefragung „Modernes Datenmanagement für die Analytik" (BARC 2015), Ergebnisstudie kostenfrei verfügbar unter *www.barc.de* im Bereich Research.

die wesentlichen Anforderungen, die in den letzten Jahren beispielsweise den Trend zu „Self Service BI“ angefeuert haben, also der Bereitstellung weitgehender Möglichkeiten zur Zusammenstellung, Aufbereitung und Visualisierung von Daten für Fachanwender. Da dies häufig auch mit einer „Self Service-Datenintegration“ verbunden ist, ergibt sich schnell die Kehrseite der Medaille solcher Initiativen: Die Konsistenz von Daten kann in einer dezentralisierten Welt individueller Datenaufbereitung – wenn überhaupt – nur mit erheblichen Anstrengungen einer Data Governance sichergestellt werden. Die ersten Unternehmen kehren demnach auch schon wieder zu stärker zentralistisch ausgerichteten Konzepten zurück, um dem Daten-Wildwuchs Einhalt zu gebieten.

Dieser Spagat zwischen der Bereitstellung qualitätsgesicherter Daten unter übergreifender Kontrolle auf der einen sowie Flexibilität und Individualität in Datenzusammenstellung und -auswertung auf der anderen Seite ist aus unserer Sicht die momentan größte Herausforderung für Betreiber entscheidungsunterstützender Informationssysteme.

Das Buch zeigt, dass viele Methoden und Technologien hierfür zur Verfügung stehen. Werden sie richtig eingesetzt, sind dem Data Warehouse auch weitere 25 Jahre erfolgreichen Einsatzes beschieden, denn Entscheidungsträger im Unternehmen werden auch in Zukunft nicht auf konsistente und qualitätsgesicherte Daten zur Entscheidungsfindung verzichten.

Würzburg, den 14.7.2016

Dr. Carsten Bange

Über dieses Buch

Das vorliegende Buch ist eine Weiterentwicklung des Buches „Data Warehousing mit Oracle – Business Intelligence in der Praxis“, das 2011 beim Carl Hanser Verlag erschienen und mittlerweile vergriffen ist. Im Vergleich zur vorherigen Version wurden hier die allgemeinen Konzepte, Architekturvorschläge und Vorgehensweisen stark ausgebaut und aktualisiert. Oracle-spezifische Informationen wurden – bis auf die Verwendung in Beispielen – weitgehend verallgemeinert, sodass die vorliegenden Blueprints auch für andere Datenbanktechnologien eingesetzt werden können.

Die Data Warehouse Blueprints wurden vorerst als interner Leitfaden für die BI-Consultants bei Trivadis zur Verfügung gestellt, bevor sie öffentlich publiziert wurden. Während dieser Zeit haben verschiedene Trivadis-Kollegen die einzelnen Kapitel überprüft und zahlreiche Korrekturen, Änderungsvorschläge und Ergänzungen zur nun vorliegenden Ausgabe beigetragen.

Die Autoren

Dani Schnider

Dani Schnider ist seit seinem abgeschlossenen Informatikstudium an der ETH Zürich (1990) in der Informatik tätig. Seit 1997 arbeitet er vorwiegend in DWH-Projekten. Konzeption, Design, Aufbau und Weiterentwicklung von Data Warehouses, logische und physische Datenmodellierung im DWH-Umfeld sowie Reviews, Architekturberatungen und Schulungen bilden seine Aufgabenschwerpunkte in diesem Bereich. Präsentationen an verschiedenen Konferenzen und Publikationen von Fachartikeln und Blog-Posts runden seine Tätigkeiten ab. (Kontakt: dani.schnider@trivadis.com)

Claus Jordan

Seit seinem Abschluss des Studiums der Wirtschaftsinformatik 1993 ist Claus Jordan im Umfeld Data Warehouse und Business Intelligence aktiv. Seit 2003 bringt er seine Erfahrung in diesen Bereichen für die Trivadis GmbH in zahlreichen Kundenprojekten, als Trainer und als Autor ein. Seine Schwerpunkte liegen dabei im Design unterschiedlicher Datenmodellierungsmethoden eines Data Warehouse, sowie in der Implementierung von ETL-Prozessen und deren Standardisierung. (Kontakt: claus.jordan@trivadis.com)

Peter Welker

Peter Welker arbeitete bereits vor dem Abschluss seines Studiums der Medizininformatik 1996 als Entwickler für Anwendungssoftware. 1998 wechselte er ins Data Warehousing und ist seitdem hauptsächlich in Projekten mit dem Fokus ETL, DWH-Lösungsarchitektur, Review und Performance aktiv. In den letzten Jahren beschäftigt er sich intensiv mit den neuen Technologien. Er präsentiert an Konferenzen, publiziert Fachartikel und verantwortet bei der Deutschen Oracle-Anwendergruppe (DOAG) das Thema „Big Data“. (Kontakt: peter.welker@trivadis.com)

Joachim Wehner

Seit seiner Diplomarbeit „Werkzeuge zum Aufbau eines Data Warehouses“ aus dem Jahre 1996 lässt ihn dieses Thema nicht mehr los. Als Berater und Trainer arbeitet Joachim Wehner über die Jahre primär in BI-/DWH-Kundenprojekten. Im Mittelpunkt stehen dabei fast immer die Architektur, das Design sowie Reviews solcher Data-Warehouse-Umgebungen. Inzwischen hat sich sein Verantwortungsbereich von der Technik auf die Managementseite verlagert. (Kontakt: joachim.wehner@trivadis.com)

Danksagung

Die Kapitel dieses Buches wurden von verschiedenen Trivadis-Consultants geprüft, korrigiert und mit wertvollen Ergänzungen und Änderungsvorschlägen angereichert. Der Dank für diese Reviewarbeit gilt folgenden Personen: Adrian Abegglen, Aron Hennerdal, Beat Flühmann, Christoph Hisserich, Kamilla Reichardt, Maurice Müller, Peter Denk, Stanislav Lando, Thomas Brunner und Willfried Färber. Die gute und konstruktive Zusammenarbeit mit Frau Hasselbach und Frau Weilhart vom Hanser Verlag ist an dieser Stelle ebenfalls dankend zu erwähnen wie das passende Geleitwort zu diesem Buch von Herrn Dr. Carsten Bange vom Business Application Research Center (BARC).

Ein besonderer Dank gilt natürlich der Trivadis AG für die Unterstützung des Buchprojekts während der Erstellung und dafür, dass sie es ermöglicht hat, dass das Buch wiederum öffentlich publiziert werden kann.

1 Einleitung

Business Intelligence (BI) und Data Warehousing (DWH) sind zwei Begriffe, die in vielen Unternehmen nicht mehr wegzudenken sind und denen eine immer wichtigere Bedeutung zukommt. In vielen großen Unternehmen gehören Data Warehouses und BI-Applikationen zu den zentralen Systemen. Auch kleinere und mittlere Betriebe benutzen Business Intelligence für die Planung und Überprüfung ihrer Geschäftsziele.

Business Intelligence bezeichnet die systematische Auswertung von Daten eines Unternehmens, um damit Geschäftsprozesse zu analysieren und zu optimieren. Das Ziel von Business Intelligence ist es, aus vergleichbaren Kennzahlen neue Erkenntnisse zu gewinnen. Sie dienen als Basis für strategische und operative Entscheidungen, mit denen die Unternehmensziele besser erreicht werden können.

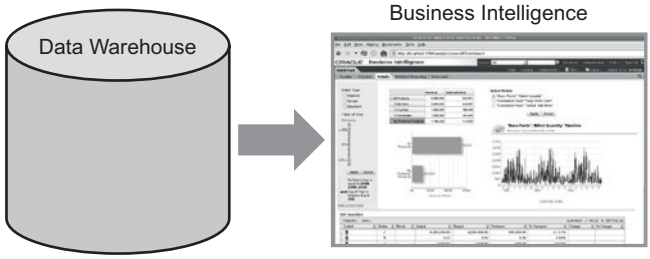

Bild 1.1 Data Warehouse als Basis für Business Intelligence

Um Business Intelligence erfolgreich betreiben zu können, muss eine solide Data-Warehouse-Architektur als Basis vorhanden sein. Ziel eines Data Warehouse ist es, die Daten aus verschiedenen operativen Quellsystemen so zusammenzuführen und in geeigneter Form abzuspeichern, dass darauf einfache und flexible Abfragen sowie verschiedenste Arten von Auswertungen möglich sind.

Data Warehouse oder Business Intelligence?

Die Begriffe „Data Warehouse“ und „Business Intelligence“ werden teilweise in unterschiedlichem Kontext verwendet. So wird „Business Intelligence“ einerseits als Sammelbegriff für BI-Gesamtlösungen – unter anderem Data Warehouses – verwendet, andererseits für die systematische Analyse von Geschäftsprozessen anhand von Kennzahlen.

Ein Data Warehouse wiederum bezeichnet zum einen ein System aus Datenbank(en) und ETL-Prozessen, das die notwendigen Informationen zur Verfügung stellt, um Business Intelligence betreiben zu können. Zum anderen wird der Begriff „Data Warehouse“ oft auch für die zentrale Integrations- und Historisierungsschicht innerhalb eines DWH-Gesamtsystems verwendet.

Eine einheitliche Namensgebung innerhalb der Informatikwelt ist kaum möglich, aber zumindest im vorliegenden Buch wurde versucht, die Begriffe einheitlich zu verwenden. Hier verstehen wir unter Business Intelligence (BI) jede Art von Anwendungen zur Datenanalyse, basierend auf einem Data Warehouse (DWH). Das Data Warehouse ist somit die technische Datenbasis für Business Intelligence. Die zentrale Schicht im DWH wird hier – zur Unterscheidung vom DWH-Gesamtsystem – als Core (oder Core Data Warehouse) bezeichnet.

Ein Data Warehouse umfasst somit die technische und fachliche Basis, die notwendig ist, um Anwendungen im Bereich Business Intelligence betreiben zu können. Integration, Skalierbarkeit und Performance sind wichtige Erfolgsfaktoren. Jede BI-Applikation und jedes Data Warehouse kann nur dann erfolgreich sein, wenn die Architektur richtig aufgebaut ist, die einzelnen Komponenten zusammenpassen und das Gesamtsystem fehlerlos konfiguriert wird.

In den Data Warehouse Blueprints werden Grundlagen und Konzepte für den Aufbau und Betrieb von Data Warehouses beschrieben. Die vorliegenden Kapitel wurden soweit möglich unabhängig von einer spezifischen Technologie beschrieben und lassen sich mit unterschiedlichen Datenbanksystemen und Softwarekomponenten umsetzen. Da jedoch die Autoren alle im Oracle-Umfeld tätig sind, schimmern teilweise technologiespezifische Detailinformationen durch. Die mit anderen Datenbanktechnologien vertrauten Leserinnen und Leser werden aufgefordert, beim Lesen der folgenden Kapitel tolerant zu sein und die technologiespezifischen Begriffe sinngemäß in ihre Nomenklatur zu übersetzen.

1.1 Ziele dieses Buches

Um leistungsfähige und stabile DWH-Systeme aufbauen und betreiben zu können, ist entsprechendes Know-how über Data Warehousing notwendig. Das vorliegende Buch gibt einen Überblick über eine typische DWH-Architektur und zeigt anhand von zahlreichen Beispielen auf, wie die einzelnen Komponenten eines Data Warehouse realisiert und betrie-

ben werden können. Der Hauptfokus liegt dabei nicht auf einer vollständigen Aufzählung der technischen Möglichkeiten – dazu stehen die Dokumentationen oder entsprechende Schulungen für die jeweilige Datenbanktechnologie zur Verfügung –, sondern darauf, wie allgemeine Technologien und Methoden in konkreten DWH-Projekten verwendet werden können. Die hier aufgezeigten Konzepte und Vorgehensweisen wurden in zahlreichen Projekten eingesetzt und – basierend auf den Erfahrungen daraus – verfeinert und erweitert.

Anhand verschiedener Tipps und Tricks aus der Praxis wird erläutert, wie die beschriebenen Methoden im Data Warehousing eingesetzt werden können. Es versteht sich von selbst, dass die hier beschriebenen Möglichkeiten keinen Anspruch auf Vollständigkeit erheben. Jedes Data Warehouse hat andere Anforderungen, Systemvorgaben und Spezialfälle, die zu berücksichtigen sind. Die hier vorgestellten Konzepte sollen jedoch als technischer Leitfaden dienen, um auch komplexe und umfangreiche Data Warehouses nach bewährtem Muster aufbauen zu können.

1.2 Struktur dieses Buches

Das vorliegende Buch ist in folgende Hauptkapitel unterteilt:

- Kapitel 1 gibt einen Überblick über Data Warehousing und Business Intelligence, die Struktur des Buches und die verwendeten Begriffe.
- Kapitel 2 beschreibt die grundlegenden *Architekturen* von Data Warehouse, BI-Anwendungen und Datenhaltung innerhalb eines DWH-Systems.
- Kapitel 3 befasst sich mit der *Datenmodellierung* im Data Warehouse und beschreibt unterschiedliche Modellierungsansätze, wie sie im DWH-Umfeld zum Einsatz kommen.
- Kapitel 4 geht auf verschiedene Aspekte der *Datenintegration* von den Quellsystemen ins Data Warehouse und beschreibt verschiedene Konzepte, die in diesem Zusammenhang verwendet werden.
- Kapitel 5 befasst sich detailliert mit dem *Design der DWH-Schichten*, basierend auf der Architektur und den Grundsätzen der Datenmodellierung und Datenintegration der vorhergehenden Kapitel.
- Kapitel 6 beschreibt verschiedene Thematiken im Zusammenhang mit dem *physischen Datenbankdesign* einer DWH-Datenbank, ohne auf spezifische Features einzelner Datenbanksysteme einzugehen.
- Kapitel 7 gibt einen kurzen Überblick über verschiedene Kategorien von *BI-Anwendungen*, wie sie typischerweise in Business-Intelligence-Plattformen mit Data Warehouses zum Einsatz kommen.
- Kapitel 8 befasst sich mit unterschiedlichen Aspekten, die beim *Betrieb* eines Data Warehouse berücksichtigt werden müssen. Dazu gehören Themen wie Release Management, Deployment, Monitoring und Migration von Data Warehouses.

1.3 Hinweis zur Anwendung dieses Buches

Als „Blueprints“ – also Baupläne – werden hier Verfahren und Methoden bezeichnet, die sich in verschiedenen DWH-Projekten bewährt haben. Das bedeutet aber nicht, dass sie die einzige oder beste Lösung für **jedes** Data Warehouse beschreiben.

Erfahrene DWH-Entwickler und BI-Consultants werden nach der Lektüre der nachfolgenden Kapitel in der Lage sein, die beschriebenen Konzepte und Praxistipps soweit sinnvoll in konkreten DWH-Projekten anzuwenden. Sie sollten aber auch die nötige Erfahrung haben, bei Bedarf zu erkennen, ob und wann von den hier beschriebenen Blueprints abzuweichen ist – sei es durch den Einsatz anderer Technologien, durch spezielle Kundenbedürfnisse oder durch eine andere Ausgangslage, als sie hier angenommen wird.

Das gilt auch für die Architektur eines Data Warehouse: Die im vorliegenden Buch verwendete Architektur beschreibt eine bewährte, aber nicht die einzig mögliche Variante, wie ein Data Warehouse auszusehen hat. Vielleicht hat das DWH bei einem spezifischen Kunden mehr oder weniger Schichten, oder es werden andere Begriffe für die einzelnen Komponenten verwendet. Auch hier gilt: Was zweckmäßig ist, kann und soll aus den Blueprints übernommen werden. Wenn es sinnvoll und begründbar ist, von den hier beschriebenen Methoden abzuweichen, ist dies durchaus erlaubt und gewünscht – sofern es der Qualität des zu bauenden Data Warehouse und der Zufriedenheit des Kunden dient.

2 Architektur

Eine gut strukturierte Architektur ist eine wichtige Voraussetzung für den erfolgreichen Einsatz von Data Warehousing und Business Intelligence. Die wichtigsten Grundsätze zur Architektur von DWH-Systemen und BI-Anwendungen sowie verschiedene Möglichkeiten zur Datenhaltung in einem Data Warehouse sind in diesem Kapitel zusammengefasst.

- Abschnitt 2.1 beschreibt die grundlegende Architektur eines Data Warehouse und stellt die verschiedenen Schichten einer DWH-Architektur vor.
- In Abschnitt 2.2 werden die wichtigsten Grundsätze zur Architektur und zum Aufbau von BI-Anwendungen erläutert.
- Abschnitt 2.3 gibt einen Überblick über verschiedene Konzepte zur Datenhaltung, wie sie für Data Warehouses zum Einsatz kommen.

2.1 Data Warehouse-Architektur

Ein Data Warehouse (DWH) stellt die technische Infrastruktur zur Verfügung, die benötigt wird, um Business Intelligence betreiben zu können. Sein Zweck ist es, Daten aus unterschiedlichen Datenquellen zu integrieren und eine historisierte Datenbasis zur Verfügung zu stellen, welche für Standard- und Ad-hoc-Reporting, OLAP[1]-Analysen, Balanced Scorecards, BI-Dashboards und weitere BI-Anwendungen eingesetzt werden kann. Ein DWH ist ein abfrageoptimiertes System, mit welchem auf eine Sammlung von historisierten Daten über einen längeren Zeitpunkt zugegriffen werden kann.

Durch diese Ausgangslage ergeben sich einige Unterschiede zwischen einem operativen System (auch OLTP[2]-System genannt) und einem Data Warehouse. Während in einem OLTP-System mehrere bis viele Anwender gleichzeitig Daten einfügen, ändern und löschen, ist dies bei einem DWH-System in der Regel nicht der Fall. Die einzigen „Anwender“, die in ein Data Warehouse schreiben, sind die ETL-Prozesse, welche Daten von den Quellsystemen ins DWH laden. Auch die Art der Abfragen ist unterschiedlich. In operativen Systemen werden typi-

[1] OLAP = Online Analytical Processing
[2] OLTP = Online Transaction Processing

scherweise spezifische Informationen in einem großen Datenbestand gesucht, beispielsweise die letzten zehn Banktransaktionen eines bestimmten Kunden. In einem Data Warehouse hingegen werden meistens Auswertungen über große Datenmengen ausgeführt und aggregiert, zum Beispiel die Summe über alle Verkäufe an alle Kunden einer bestimmten Region.

Um diesen unterschiedlichen Bedürfnissen gerecht zu werden, werden DWH-Datenbanken anders aufgebaut als OLTP-Datenbanken. Architektur, Design und Datenmodellierung funktionieren im DWH-Umfeld nicht auf die gleiche Weise, wie es viele erfahrene Architekten, Datenmodellierer und Entwickler gewohnt sind, die hauptsächlich im Bereich von OLTP-Datenbanken tätig sind. Auf die spezifischen Bedürfnisse von DWH-Systemen wird deshalb nachfolgend eingegangen.

Die Komplexität und Erweiterbarkeit eines Data Warehouse ist weitgehend abhängig von der verwendeten Architektur. Deshalb ist es in jedem DWH-Projekt von Anfang an wichtig, dass eine saubere Architektur definiert und implementiert wird. In der Regel bedeutet das, dass die Architektur aus unterschiedlichen Schichten besteht. Diese Schichten decken jeweils unterschiedliche Anforderungen ab. Auch wenn dies zum Beginn des Projektes nach Mehraufwand aussieht, zahlt sich eine konsequente Aufteilung in verschiedene DWH-Schichten im späteren Projektverlauf und im operativen Betrieb des Systems aus.

Leider wird oft der Fehler gemacht, dass aufgrund von knappen Terminvorgaben wesentliche Architekturgrundsätze missachtet und „Abkürzungen“ bzw. „Schnellschüsse“ implementiert werden. Diese Ad-hoc-Lösungen können früher oder später zu Problemen führen. Der Aufwand, diese wiederum zu beheben, ist oft größer als der Aufwand, von Anfang an eine saubere Lösung zu realisieren.

2.1.1 Aufbau eines Data Warehouse

Ein Data Warehouse besteht typischerweise aus verschiedenen Schichten (auch Layers, Bereiche oder Komponenten genannt) und Datenflüssen zwischen diesen Schichten. Auch wenn nicht jedes DWH-System alle Schichten umfassen muss, lässt sich jedes Data Warehouse auf eine Grundarchitektur, wie sie in Bild 2.1 dargestellt ist, zurückführen.

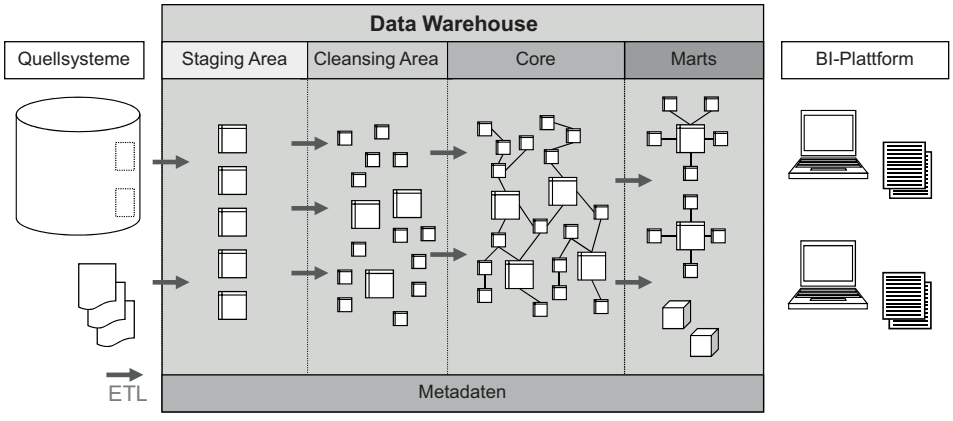

Bild 2.1 Grundarchitektur eines Data Warehouse

Um den Zweck der einzelnen Schichten in einer DWH-Architektur zu erklären, werden nachfolgend Beispiele aus dem „realen Leben" gezeigt. Nehmen wir an, das DWH sei ein großes Lebensmittelgeschäft. Auch dort gibt es verschiedene Bereiche, die jeweils einem bestimmten Zweck dienen.

Folgende Schichten oder Bereiche gehören zu einer vollständigen DWH-Architektur:

- *Staging Area:* Daten aus unterschiedlichen Quellsystemen werden zuerst in die Staging Area geladen. In diesem ersten Bereich des DWH werden die Daten so gespeichert, wie sie angeliefert werden. Die Struktur der Stage-Tabellen entspricht deshalb der Schnittstelle zum Quellsystem[3]. Beziehungen zwischen den einzelnen Tabellen bestehen keine. Jede Tabelle enthält die Daten der letzten Lieferung, welche vor der nächsten Lieferung gelöscht werden.

  In einem Lebensmittelgeschäft entspricht die Staging Area der Laderampe, an der die Lieferanten (Quellsysteme) ihre Waren (Daten) abliefern. Auch dort werden immer nur die neuesten Lieferungen zwischengelagert, bevor sie in den nächsten Bereich überführt werden.

- *Cleansing*[4] *Area:* Bevor die gelieferten Daten ins Core geladen werden, müssen sie bereinigt werden. Fehlerhafte Daten müssen entweder ausgefiltert, korrigiert oder durch Singletons (Defaultwerte) ergänzt werden. Daten aus unterschiedlichen Quellsystemen müssen in eine vereinheitlichte Form transformiert und integriert werden. Die meisten dieser Bereinigungsschritte werden in der Cleansing Area durchgeführt. Auch diese Schicht enthält nur die Daten der letzten Lieferung.

  Im Lebensmittelgeschäft kann die Cleansing Area mit dem Bereich verglichen werden, in dem die Waren für den Verkauf kommissioniert werden. Die Waren werden ausgepackt, Gemüse und Salat werden gewaschen, das Fleisch portioniert, ggf. mehrere Produkte zusammengefasst und alles mit Preisetiketten versehen. Die Qualitätskontrolle der angelieferten Ware gehört ebenfalls in diesen Bereich.

- *Core*: Die Daten aus den verschiedenen Quellsystemen werden über die Staging und Cleansing Area in einem zentralen Bereich, dem Core, zusammengeführt und dort über einen längeren Zeitraum, oft mehrere Jahre, gespeichert. Eine Hauptaufgabe des Core ist es, die Daten aus den unterschiedlichen Quellen zu integrieren und nicht mehr getrennt nach Herkunft, sondern themenspezifisch strukturiert zu speichern. Oft spricht man bei thematischen Teilbereichen im Core von „Subject Areas". Die Daten werden im Core so abgelegt, dass historische Daten zu jedem späteren Zeitpunkt ermittelt werden können. Das Core sollte die einzige Datenquelle für die Data Marts sein. Direkte Zugriffe von Benutzern auf das Core sollten möglichst vermieden werden.

  Das Core kann mit einem Hochregallager verglichen werden. Waren werden so abgelegt, dass sie jederzeit auffindbar sind, aber der Zugriff darauf ist nur internen Mitarbeitern möglich. Kunden haben im Lager nichts zu suchen – außer vielleicht bei IKEA. Im Gegensatz zu einem Hochregallager bleiben die Daten aber auch dann im Core erhalten, nachdem sie an die Data Marts übertragen wurden.

---

[3] Oft werden den Stage-Tabellen zusätzliche Attribute für Auditinformationen zugefügt, die im Quellsystem nicht vorhanden sind.

[4] Der Begriff „Cleansing" wird englisch „klensing" ausgesprochen und nicht „kliinsing".

- *Marts:* In den Data Marts werden Teilmengen der Daten aus dem Core so aufbereitet abgespeichert, *dass* sie in einer für die Benutzerabfragen geeigneten Form zur Verfügung stehen. Jeder Data Mart sollte nur die für die jeweilige Anwendung relevanten Daten bzw. eine spezielle Sicht auf die Daten enthalten. Das bedeutet, dass typischerweise mehrere Data Marts für unterschiedliche Benutzergruppen und BI-Anwendungen definiert werden. Dadurch kann die Komplexität der Abfragen reduziert werden. Das erhöht die Akzeptanz des DWH-Systems bei den Benutzern.

  Die Data Marts sind die Marktstände oder Verkaufsgestelle im Lebensmittelgeschäft. Jeder Marktstand bietet eine bestimmte Auswahl von Waren an, z. B. Gemüse, Fleisch oder Käse. Die Waren werden so präsentiert, dass sie von der jeweiligen Kundengruppe akzeptiert, also gekauft werden.

- *ETL-Prozesse:* Die Daten, die von den Quellsystemen als Files, Schnittstellentabellen oder über einen View Layer zur Verfügung gestellt werden, werden in die Staging Area geladen, in der Cleansing Area bereinigt und dann im Core integriert und historisiert. Vom Core werden aufgrund von fachlichen Anforderungen Teilmengen oder oft auch nur Aggregate in die verschiedenen Data Marts geladen.

  All diese Datenflüsse werden unter dem Begriff ETL (Extraction, Transformation, Loading) zusammengefasst. Die Extraktion der Daten aus den Quellsystemen findet in der Regel außerhalb des DWH-Systems statt, nämlich in den Quellsystemen selbst. Als Transformationen werden alle Datenumformungen, Bereinigungen, Anreicherungen mit Zusatzinformationen und Aggregationen bezeichnet. Schließlich werden die Daten in die Zieltabellen der nächsten Schicht geladen.

  Die ETL-Prozesse sind die Mitarbeiter des Lebensmittelgeschäfts, die unterschiedliche Arbeiten verrichten müssen, damit die Lebensmittel vom Lieferanten bis hin zum Kunden gelangen.

- *Metadaten*: Für den reibungsfreien Betrieb des Data Warehouse werden unterschiedliche Arten von Metadaten benötigt. Fachliche Metadaten enthalten fachliche Beschreibungen aller Attribute, Drill-Pfade und Aggregationsregeln für die Frontend-Applikationen und Codebezeichnungen. Technische Metadaten beschreiben z. B. Datenstrukturen, Mapping-Regeln und Parameter zur ETL-Steuerung. Operative Metadaten beinhalten alle Log-Tabellen, Fehlermeldungen, Protokollierungen der ETL-Prozesse und vieles mehr. Die Metadaten bilden die Infrastruktur eines DWH-Systems und werden als „Daten über Daten“ beschrieben.

  Auch in unserem Lebensmittelgeschäft braucht es eine funktionierende Infrastruktur – von Wegweisern zur Kasse bis hin zur Klimaüberwachung der Frischwaren.

Nicht jedes Data Warehouse hat genau diesen Aufbau. Teilweise werden einzelne Bereiche zusammengefasst – zum Beispiel Staging Area und Cleansing Area – oder anders bezeichnet. So wird zum Teil das Core als „Integration Layer“ oder als „(Core) Data Warehouse“ bezeichnet. Wichtig ist jedoch, dass das Gesamtsystem in verschiedene Bereiche unterteilt wird, um die unterschiedlichen Aufgabenbereiche wie Datenbereinigung, Integration, Historisierung und Benutzerabfragen zu entkoppeln. Auf diese Weise kann die Komplexität der Transformationsschritte zwischen den einzelnen Schichten reduziert werden.

2.1.2 Transformationsschritte

Mithilfe der ETL-Prozesse werden die Daten von den Quellsystemen ins Data Warehouse geladen. In jeder DWH-Schicht werden dabei unterschiedliche Transformationsschritte durchgeführt, wie in Bild 2.2 dargestellt.

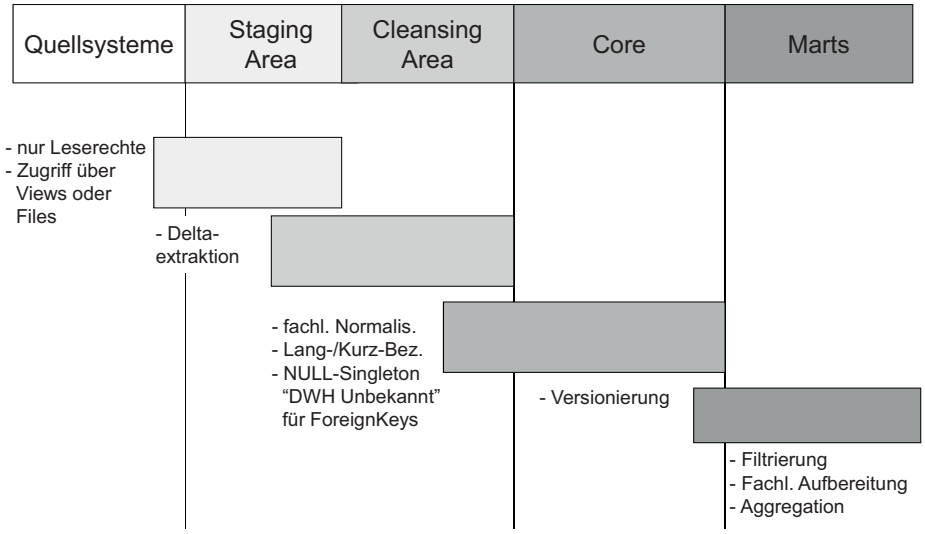

Bild 2.2 Transformationsschritte im Data Warehouse

- *Quellsystem → Staging Area*
  Wird direkt auf ein relationales Quellsystem zugegriffen, sollte als Schnittstelle zwischen Quellsystem und DWH ein View Layer als definierte Zugriffsschicht implementiert werden. Der Zugriff der ETL-Prozesse auf das Quellsystem erfolgt dann ausschließlich über diese Views. Auf diese Weise kann eine gewisse Unabhängigkeit der ETL-Prozesse gegenüber Strukturänderungen auf dem Quellsystem erreicht werden. Die Views können außerdem für die Delta-Extraktion verwendet werden, indem sie so implementiert werden, dass nur die jeweils relevante Teilmenge der Daten in den Views zur Verfügung steht. Dieses Verfahren wird teilweise für Change Data Capture (CDC) verwendet.

  Als Alternative zum direkten Zugriff auf das Quellsystem werden häufig Dateien als Schnittstelle zwischen Quellsystem und Data Warehouse verwendet. Die Extraktion der Daten in die Dateien erfolgt auf dem Quellsystem und wird meistens außerhalb des DWH-Projektes realisiert.

  Die Daten, ob über Views oder Files geliefert, werden unverändert in die Staging Area geladen und ggf. mit Auditinformationen angereichert.
- *Staging Area → Cleansing Area*
  Beim Laden in die Cleansing Area werden die Daten geprüft, bereinigt und mit zusätzlichen Attributen angereichert. Dazu gehört zum Beispiel die Ermittlung von Lang- und Kurztexten aus den fachlichen Attributen der Quellsysteme. Fehlende oder fehlerhafte Attribute und Foreign Keys werden durch Singletons ersetzt.

Fehlerhafte Datensätze können je nach Anforderungen ignoriert, aufgrund von fixen Regeln korrigiert, durch Singletons ersetzt oder in Fehlertabellen geschrieben werden. Fehlertabellen können als Basis für Fehlerprotokolle oder manuelle Korrekturen verwendet werden. Bei solchen aufwendigen Varianten der Fehlerbehandlung muss allerdings organisatorisch geklärt werden, wer für die Fehlerkorrekturen verantwortlich ist.

- *Cleansing Area → Core*
  Nachdem die Daten in der Cleansing Area in die benötigte Form aufbereitet wurden, werden sie ins Core geladen. In diesem Schritt findet die Versionierung der Stammdaten[5] statt, d. h., es wird für jeden Datensatz geprüft, ob sich etwas geändert hat und somit eine neue Version erstellt werden muss. Je nach Historisierungsanforderungen und Core-Datenmodell gibt es verschiedene Varianten der Versionierung von Stammdaten.

  Die Bewegungsdaten[6] werden historisiert. Weil sich Bewegungsdaten nachträglich nicht mehr ändern, heißt das, dass laufend neue Daten eingefügt und über einen längeren Zeitraum gespeichert werden. Oft besteht die Anforderung, dass Bewegungsdaten nach einer gewissen Zeit – in der Regel nach mehreren Jahren – aus dem Core gelöscht werden.

  Aggregationen werden im Core nicht durchgeführt. Die Bewegungsdaten werden auf der Detaillierungsstufe, die geliefert wird, gespeichert.
- *Core → Marts*
  Die Transformationen vom Core in die Data Marts bestehen aus der Filtrierung der Daten auf die für jeden Data Mart erforderliche Teilmenge, der fachlichen Aufbereitung der Dimensionen[7] in die gewünschten Hierarchiestufen sowie – falls erforderlich – der Aggregation der Bewegungsdaten auf die Granularität der Faktentabellen.

2.1.3 Architekturgrundsätze

Obwohl sich die Architektur vieler DWH-Systeme in Details unterscheidet und oft auch unterschiedliche Namen für die einzelnen Bereiche verwendet werden, gibt es ein paar wichtige Architekturgrundsätze, die auf jeden Fall berücksichtigt werden sollten. Vereinfachungen der Architektur sind erlaubt, aber die wichtigsten Schichten sollten auf keinen Fall weggelassen werden.

---

[5] Stammdaten (oder Referenzdaten) sind zustandsorientierte Daten, die sich im Laufe der Zeit ändern können. Um die Änderungen im Core nachvollziehbar abspeichern zu können, werden die Daten versioniert. Das heißt, dass für jede Datenänderung ein neuer Datensatz im Core eingefügt wird. Die Versionierung von Stammdaten wird in Kapitel 3 (Datenmodellierung) genau erklärt.

[6] Bewegungsdaten (oder Transaktionsdaten) sind ereignisorientierte Daten, die aufgrund eines bestimmten Ereignisses (z. B. Transaktion, Messung) entstehen und nachträglich nicht mehr geändert werden. Sie sind immer mit einem Ereigniszeitpunkt (z. B. Transaktionsdatum) verbunden. Die Historisierung von Bewegungsdaten wird in Kapitel 3 (Datenmodellierung) beschrieben.

[7] Die Begriffe Dimensionen, Hierarchien und Fakten sind Elemente der dimensionalen Modellierung und werden in Kapitel 3 (Datenmodellierung) erläutert.