Business News

DOAG Zeitschrift für die Anwender von Oracle Business- und BI-Lösungen

BS

Herbst 2018 | ISSN 09 36-0360 | http://bs.doag.org


ORAWORLD
Das e-Magazine für alle Oracle-Anwender!
EOUC
EMEA ORACLE USERGROUP COMMUNITY
• Spannende Geschichten aus der Oracle-Welt
• Technologische Hintergrundartikel
• Leben und Arbeiten heute und morgen
• Einblicke in andere User Groups weltweit
• Neues (und Altes) aus der Welt der Nerds
• Comics, Fun Facts und Infografiken
Jetzt Artikel einreichen oder Thema vorschlagen!
Jetzt e-Magazine herunterladen
www.oraworld.org

Sven Bosinger
DOAG-Themenverantwortlicher
Data Analytics

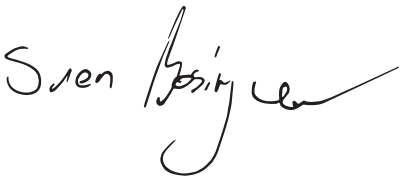
Liebe Mitglieder, liebe Leserinnen und Leser,
im März 2018 fand die diesjährige DOAG Data Analytics Konferenz im Phantasialand in Brühl statt. Der Eindruck, den diese Konferenz bei mir hinterlassen hat, lässt sich am besten beschreiben mit bunt, spannend, innovativ und kreativ. Es hat sich wieder gezeigt, dass Data Analytics in seiner gesamten Spannbreite eine unglaubliche Themenvielfalt beinhaltet. Highlights für mich waren Themen wie das neue Oracle Autonomous Data Warehouse, revisionssichere Datenqualität, agile Data-Warehouse-Projekte, eine Podiumsdiskussion zur künstlichen Intelligenz und vieles mehr.

Auch nach vielen Jahren ist die Innovationskraft sowohl der Hersteller als auch der Anwender ungebrochen. Es gibt kein Jahr, in dem nicht wegweisende und wichtige Projekte die Grenze des Machbaren wieder um einen Schritt verschieben.

Das Portfolio Data Analytics ist immer auch ein Spiegel dessen, was in unserer Gesellschaft und den agierenden Unternehmen aktuell passiert. So finden sich hier Projekte zur Autonomisierung von Systemen genauso wie zur Vorhersage von Wartungsintervallen oder zur Visualisierung von Informationen in interaktiven Karten. Data Analytics ist genauso vielfältig wie das wahre Leben mit all seinen Facetten.

Auch in dem vorliegenden Magazin präsentieren wir Ihnen die volle Bandbreite dessen, was Data Analytics ausmacht. Viel Spaß beim Eintauchen in die große Vielfallt der Themen.

Ihr



Impressum

DOAG Business News wird von der DOAG Deutsche ORACLE-Anwendergruppe e.V. (Tempelhofer Weg 64, 12347 Berlin, www.doag.org), herausgegeben. Es ist das User-Magazin rund um die Applikations-Produkte der Oracle Corp., USA, im Raum Deutschland, Österreich und Schweiz. Es ist unabhängig von Oracle und vertritt weder direkt noch indirekt deren wirtschaftliche Interessen. Vielmehr vertritt es die Interessen der Anwender an den Themen rund um die ORACLE-Produkte, fördert den Wissensaustausch zwischen den Lesern und informiert über neue Produkte und Technologien.

DOAG Business News wird verlegt von der DOAG Dienstleistungen GmbH, Tempelhofer Weg 64, 12347 Berlin, Deutschland, gesetzlich vertreten durch den Geschäftsführer Fried Saacke, deren Unternehmensgegenstand Vereinsmanagement, Veranstaltungsorganisation und Publishing ist.

Die DOAG Deutsche Oracle-Anwendergruppe e.V. hält 100 Prozent der Stammeinlage der DOAG Dienstleistungen GmbH. Die DOAG Deutsche Oracle-Anwendergruppe e.V. wird gesetzlich durch den Vorstand vertreten; Vorsitzender: Stefan Kinnen. Die DOAG Deutsche Oracle-Anwendergruppe e.V. informiert kompetent über alle Oracle-Themen, setzt sich für die Interessen der Mitglieder ein und führen einen konstruktiv-kritischen Dialog mit Oracle.

Redaktion:
Sitz: DOAG Dienstleistungen GmbH
(Anschrift s.o.)
Chefredakteur (ViSdP): Wolfgang Taschner
Kontakt: redaktion@doag.org
Weitere Redakteure: Lisa Damerow, Mylène Diacquenod, Marina Fischer, Sanela Lukavica, Martin Meyer, Fried Saacke, Rolf Scheuch, Dr. Frank Schönthaler

Druck:
adame Advertising and Media GmbH, Berlin, www.adame.de

Fotonachweis:
Titel: © spainter_vfx/123RF
S. 5: © 董 豆豆/123RF
S. 9: © Kheng Ho Toh/123RF
S. 14: © Kanda Euatham/123RF
S. 19: © aimage/123RF
S. 24: © Anton Balazh/123RF
S. 28: © Artisticco LLC/123RF

Titel, Gestaltung und Satz:
Caroline Sengpiel,
DOAG Dienstleistungen GmbH
(Anschrift s.o.)

Anzeigen:
Simone Fischer, DOAG Dienstleistungen GmbH
(verantwortlich, Anschrift s.o.)
Kontakt: anzeigen@doag.org

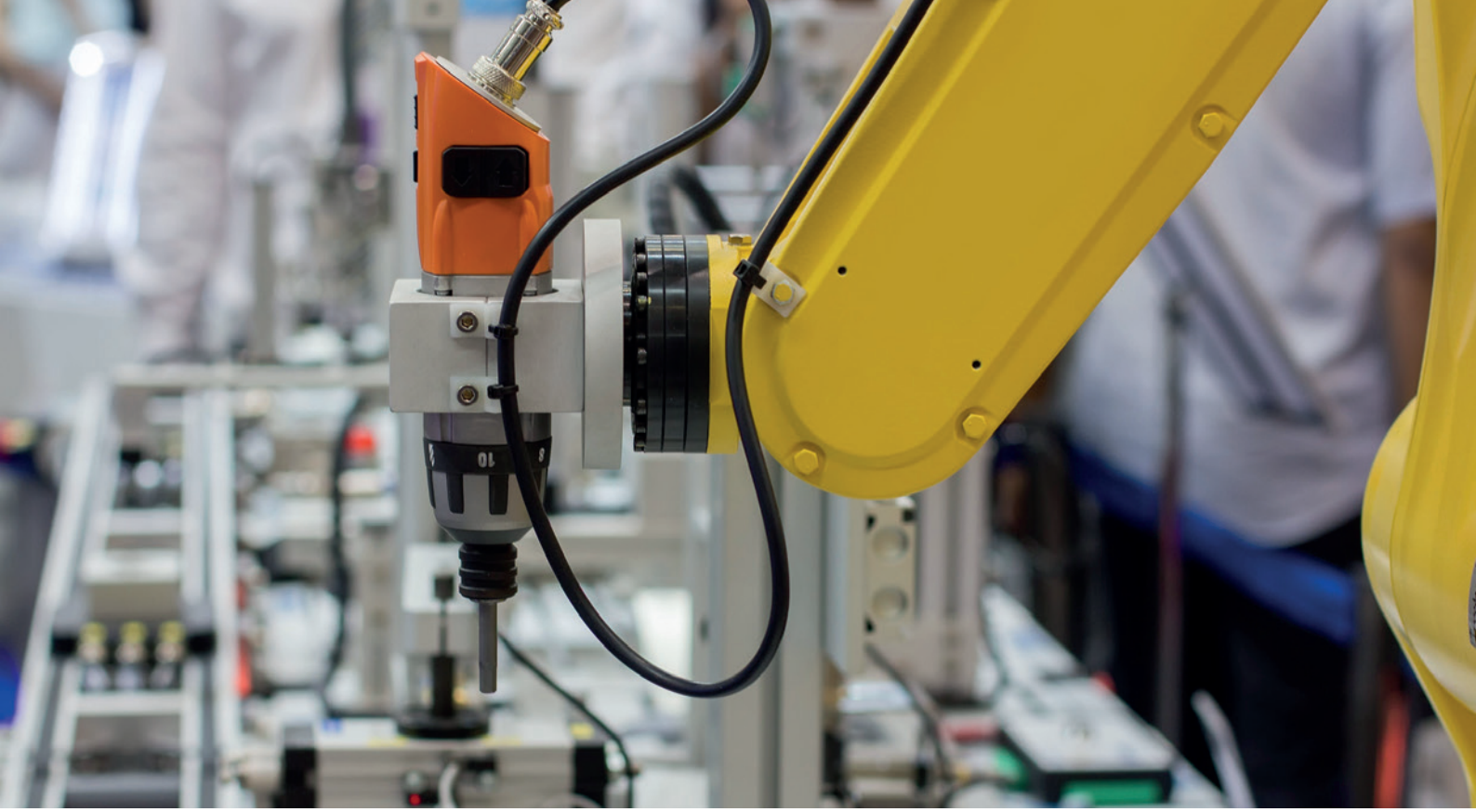
Mediadaten und Preise unter: www.doag.org/go/mediadaten

Data Vault hat sich als Modellierungsverfahren für Data Warehouses etabliert

Reine Präsentation von Daten ohne visuelle Aufbereitung ist heute kaum vorstellbar

Immer mehr BI-Manager werden beim Thema „Datenqualität" in die Verantwortung genommen

Unsere Inserenten



BI-Strategie und Industrialisierung

Hartmut Westenberger, Technische Hochschule Köln

Über Jahre hinweg evolutionär gewachsene Business-Intelligence-Infrastrukturen erweisen sich zunehmend als wenig transparent und hochkomplex, damit also schwer wart- und erweiterbar. Um der fortschreitenden Digitalisierung und dem Bedarf nach einer zeitnahen Informationsversorgung gerecht zu werden, sind Ansätze zur Beherrschung der zunehmenden Komplexität zu betrachten, indem alle BI-Architektur- und Lösungsbausteine abhängig von ihrem Reifegrad beziehungsweise von ihrem Beitrag zur Wettbewerbs-Differenzierung auf Standardisierbarkeit geprüft werden. Komplexitätsreduzierend können ebenfalls die Dekomposition der BI-Prozesse und die Automatisierung einzelner Arbeitsschritte oder die Auslagerung an externe Dienstleistung wirken. Empirische Studien geben Einblick, in welchem Umfang diese Industrialisierungsmuster in BI-Strategien einbezogen werden.

Business Intelligence (BI) als die Fähigkeit einer organisatorischen Einheit, entscheidungsrelevante Informationen zu ermitteln, bedarfsgerecht zur Verfügung zu stellen und in Steuerungsprozesse einzubeziehen, steht gegenwärtig vor großen Herausforderungen. Die Dynamisierung und Digitalisierung der Geschäftsmodelle sowie neue Formen von Datenquellen und Technologien wirken als Treiber für eine erhebliche Ausweitung des Informationsbedarfs.

Damit wird die BI-Infrastruktur vielfältiger. Neben den ERP-Systemen als relationalen Datenquellen kommen externe Daten sowie polystrukturierte Daten (Big Data) hinzu. Es erfolgt ein Wandel von einem zentralen Enterprise Data Warehouse (EDW) zu einem Data Ecosystem. Dies kann neben dem Data Warehouse (DWH) und den Data Marts (DM) auch Spiegelungen der operativen Datenquellen (ODS, Operational Datastore) und ein Data Lake sowie ein Metadata-Repository mit den entsprechenden syntaktischen und semantischen Informationen enthalten *(siehe Abbildung 1)*.

Reporting, Analyse und Planung werden durch weitere BI-Anwendungen wie BI-Sandboxes ergänzt, die dem Nutzer definierte Freiräume für die Einarbeitung eigener Datenquellen gewähren, oder wie dem Event-Processing, in dem Informationen aus der BI-Infrastruktur für eine automatisierte Prozesssteuerung benutzt werden. Erweitert wird die dedizierte BI-Infrastruktur durch „Embedded BI“ (in operativen Anwendungen integrierte BI-Funktionalitäten) sowie „Shadow BI“.

Die biMA-Studie 2012/13 benennt die unzureichende Datenqualität, die oft fehlende BI-Strategie und BI-Governance sowie die mangelnde Flexibilität als die drei zentralen Herausforderungen im BI-Umfeld [1]. Dieser Befund hat sich im Verlauf der letzten Jahre nicht wesentlich geändert, wie die biMA-Studie 2017/18 beweist. Denn der Anteil der Nennungen für die Problemfelder „unzureichende Datenqualität“, „keine allgemein akzeptierte BI&Analytics-Strategie“ und „hohe Komplexität der BI-Systemlandschaft“ ist im Vergleich zur Studie 2012/13 sogar angestiegen [2].

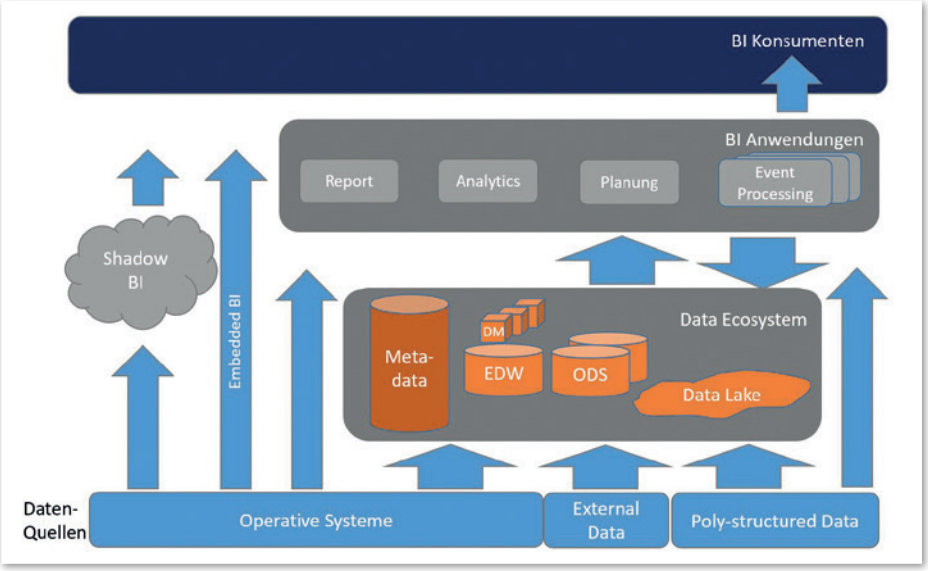


Abbildung 1: Exemplarische Architektur-Bausteine einer BI-Infrastruktur

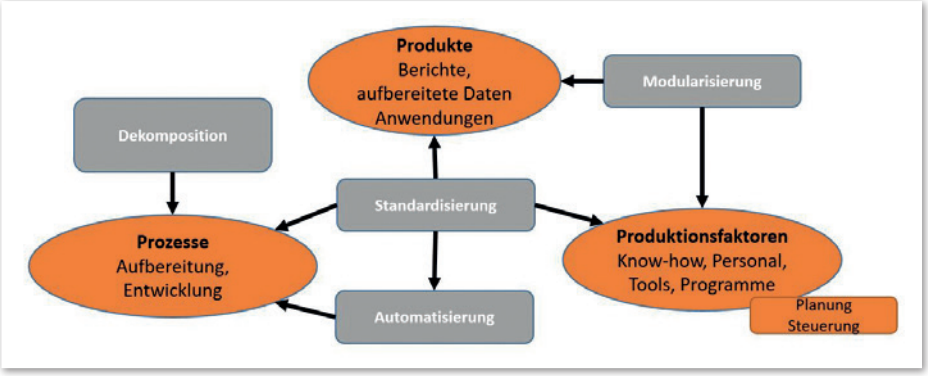


Abbildung 2: Wirkungsmuster der Industrialisierung in der BI-Factory

Dazu passen auch Ergebnisse einer Umfrage der PPI AG in Kooperation mit der TDWI Germany e.V. Sie identifiziert die mangelnde Flexibilität bei der Umsetzung und den hohen Aufwand bei der Weiterentwicklung als die häufigsten Data-Warehouse-Defizite [3]. Zugleich wird auf die oft fehlende DWH-Gesamtstrategie und die unzureichende Governance hingewiesen. Dieser Studie zufolge setzen sich viele Unternehmen mit der Frage des Neubaus beziehungsweise der Sanierung ihres EDW auseinander. Dabei wird die Erhöhung der Flexibilität als erstes Ziel des Umbaus, gefolgt von dem Ziel der Qualitätssteigerung, genannt.

Im BI Trend Monitor des Forschungsinstituts BARC belegt Master Data/Data Quality Management im Ranking der Top-BI-Trends Platz 1 [4]. Self-Service BI und Data Governance folgen auf den Plätzen 3 und 4 sowie DWH Modernization auf Platz 7. Die Autoren der Studie sehen IT-Abteilungen mit sich schnell ändernden analytischen Anforderungen konfrontiert, wobei sie im Wettbewerb mit neuen, kostengünstigen Angeboten externer Dienstanbieter stehen oder zumindest kollaborative Ansätze verfolgen sollten [4, Seite 31].

Zusammenfassend lässt sich festhalten, dass Business Intelligence in Unternehmen vielfältige Herausforderungen zu bewältigen hat, wie die Verbesserung der Datenqualität, die Beherrschung der Systemkomplexität, die Steuerung des Maßes der Selbstbestimmung der Informationsverbraucher (Self Service BI, Standardisierung), die Gestaltung der BI-Rollen und Zuständigkeiten, die Einbeziehung externer Dienste (BIaaS, Cloud und Make-or-buy) und die Einsetzbarkeit neuer Technologien. Daraus ergeben sich verschiedene Fragen wie zum Beispiel: „Können die gegensätzlichen Zielsetzungen Agilität und Qualität aus einem System heraus zugleich erfüllt werden oder sind dafür zwei weitgehend getrennte Systeme nötig?“, „Ist eine Arbeitsteilung bei der Entwicklung der DWH/BI-Infrastruktur und die Reduktion der Implementierungstiefe von DWH/BI-Infrastruktur angezeigt, zum Beispiel durch den Einsatz von vorkonfigurierten Standardkomponenten oder Standardgesamtlösungen?“, „Unter welchen Bedingungen ist es sinnvoll, den Betrieb der DWH/BI-Infrastruktur teilweise oder komplett auszulagern oder zumindest in Teilen zu automatisieren?“, „Welche Rolle spielen dabei Cloud-Lösungen?“ und „Für welche Anwender bietet BI-as-a-Service interessante Perspektiven?“ – Aspekte, die im Rahmen einer BI-Strategie behandelt und durch eine BI-Governance gesteuert werden sollten.

BI-Industrialisierung

Die IT-Strategieentwicklung hat in der Vergangenheit wesentliche Impulse aus der Betrachtung historischer Entwicklungsmuster wie der Industrialisierung erfahren [5, 6]. In der analogen Betrachtung von Business Intelligence wird die BI-Systemstruktur zur BI-Factory; die BI-Produkte sind je nach Betrachtung die Informationen am Ende der Aufbereitungskette oder BI-Anwendungen selbst.

BI-Zwischenprodukte sind die aufbereiteten Daten im Data Ecosystem, Prozesse werden durch die Datenaufbereitung, Berichtsgenerierung und die zugehörigen Entwicklungsprozesse abgebildet. Know-how, Personal, Werkzeuge und Programme bilden die wesentlichen Produktionsfaktoren. Darüber hinaus werden Planung und Steuerung des Systems als dispositiver Faktor unter den Produktionsfaktoren subsumiert *(siehe Abbildung 2)*.

Wesentliches Merkmal der Industrialisierung ist der Effizienzgewinn beim Einsatz des Produktionsfaktors „menschliche Arbeit“ durch Spezialisierung in der Wertschöpfungskette (Arbeitsteilung durch Dekomposition der Prozesse und Modularisierung der Produkte) auf der Basis einer Teil-Standardisierung der End- und Zwischenprodukte sowie der Arbeitsschritte. Die Standardisierung ermöglicht einerseits eine Reduktion der Produktionstiefe (Einsatz von Standard-Software oder Auslagerung entsprechender Funktionen), andererseits durch die damit verbundene hohe Wiederholrate der Produktionsschritte den wirtschaftlichen Einsatz von Maschinen zur Automatisierung einzelner Wertschöpfungsschritte.

Die folgenden Standpunkte zur Einsetzbarkeit der Industrialisierungsansätze basieren auf den genannten empirischen Studien sowie auf Umfragen und Interviews des Autors. Sie geben subjektive Meinungen ohne Anspruch auf Allgemeingültigkeit wieder und besitzen hypothetischen Charakter.

Dekomposition und Sourcing

Die Einbeziehung externer Dienstleister in die Onshore-Entwicklung und den Onshore-Betrieb der BI-Systemlandschaft scheint in den meisten Unternehmen etabliert. Einerseits wird damit eine Flexibilisierung des Engpasses Personal erwirkt, andererseits können spezialisierte Dienstleister den kostenintensiven Aufbau des Know-hows BI-spezifischer Entwicklungswerkzeuge wirtschaftlicher leisten. Die kooperative Entwicklung kann zudem zum Insourcing von Technologie-Know-how genutzt werden, was insbesondere bei neuen Technologien ein wichtiger strategischer Aspekt sein kann.

Die in BI-Infrastrukturen dominierende Schichten-Architektur ermöglicht eine Arbeitsteilung durch die Trennung der Datenbewirtschaftung von den BI-Anwendungen. Innerhalb der Datenbewirtschaftung selbst lassen sich einzelne Funktionen wie der Beladeprozess des Data Warehouse kapseln und an externe Dienstleister auslagern (Outsourcing). Den offensichtlichen Vorteilen hinsichtlich Flexibilisierung und Zeitgewinn sind die Aufwände und Risiken des Betriebs einer Schnittstelle zur Dienstauslagerung gegenüberzustellen. Dabei geht es neben der Wirtschaftlichkeit auch um die Aspekte „Qualität", „Sicherheit" und manchmal auch um „Wettbewerbs-Differenzierung". Eine differenzierende Sourcing-Strategie wird Technik-geprägte und standardisierbare Aufgaben eher auslagern als anwendungsnahe und unmittelbar geschäftserfolgsrelevante Dienste.

Zum Thema „Offshoring" bestehen kontroverse Standpunkte. Unklare Anforderungsspezifikationen, fehlendes Branchenwissen des Dienstleisters, unzureichende Zusammenarbeit, kulturelle Unterschiede, fehlende Vertrauensbasis sowie eine Verlängerung der Prozesswege mit dem Resultat einer reduzierten Umsetzungsgeschwindigkeit von Anforderungen werden häufig als Problemquellen genannt und haben zu zahlreichen negativen Erfahrungen beziehungsweise zu einer Skepsis gegenüber der Sinnhaftigkeit von Offshoring geführt.

Andere Unternehmen praktizieren diese Form der Arbeitsteilung bereits seit längerer Zeit erfolgreich, indem sie Know-how für das Management der externen Dienste aufgebaut und die Prozesse an der Schnittstelle zum Dienstleister optimiert haben. IT-Abteilungen, die gemäß Rogers Diffusionsmodell der „Late Majority" zuzurechnen sind, verhalten sich eher abwartend, solange entsprechende Best Practices nicht etabliert sind.

Eine Alternative zum Outsourcing ist die Nutzung von BI-Diensten in einer externen Cloud (BI-as-a-Service, BIaaS), was vor allem Vorteile hinsichtlich der Agilität bietet [7]. Obwohl viele Anbieter von BI-Infrastruktur-Lösungen diese Nutzungsform intensiv bewerben und Vorteile in Bezug auf Zeitvorteile und Flexibilität naheliegend sind, herrscht bei vielen Anwendern deutliche Zurückhaltung. So erscheint Cloud BI/BIaaS im Ranking der zwanzig wichtigsten BI-Trends des BI Trend Monitors 2018 auf einem der letzten Plätze [4, Seite 13]. Vor allem der Kontrollverlust über die Daten, regulatorische Anforderungen sowie Kostenvorbehalte werden als Argumente gegen die Cloud-Nutzung vorgebracht. Trotzdem scheint die Meinung vorzuherrschen, dass sich Cloud-Dienste in Zukunft durchsetzen werden.

Anbieter von Cloud-Diensten reagieren auf Sicherheitsbedenken, indem sie Dienste in regional angesiedelten Rechenzentren betreiben. Der Cloud-Einsatz wurde oft von Fachabteilungen vorangetrieben, in denen Daten nicht die höchste Schutzbedürftigkeit haben. Allerdings ist in einigen Unternehmen die Nutzung von Cloud-Diensten bereits fester Bestandteil der IT-Strategie geworden. In den Umfragen des Autors sieht eine deutliche Mehrheit eine stark wachsende Bedeutung von BIaaS.

Standardisierung

Standardisierung kann als das Bestreben einer Organisation zur Vereinheitlichung der Entitäten einer Domäne aufgefasst werden. Das Maß der Vereinheitlichung ist im Rahmen der Governance zu beschreiben und reicht von zwingender Vorgabe bis zur Empfehlung von Good/Best Practices, Templates sowie Referenz-Modellen für Methoden, Prozesse, Werkzeuge, Modelle, Datenstrukturen oder Anwendungs-Software. Im BI-Bereich bietet sich ein breites Spektrum für die Standardisierung, beispielsweise für die Visualisierung, die Auswahl von Architektur- und Lösungs-Bausteinen inklusive Entwicklungs-, Belieferungs- und Dokumentations-Prozessen sowie für Datenstrukturen und Technologien. Auch Templates für die Anforderungsermittlung oder Glossare fallen darunter.

Branchenspezifische BI-Standardlösungen haben sich nicht im selben Umfang durchgesetzt wie bei ERP-Lösungen. Dies wird auf die Breite und Vielfältigkeit der Anforderungen, die Vielzahl der zu vernetzenden Quell- und Ziel-Systeme sowie auf die

hohen Kosten von BI-Standard-Systemen zurückgeführt. Allerdings liegt für Unternehmen, die Software-Lösungen eines Generalisten wie SAP im operativen Bereich bevorzugen, die Nutzung der entsprechenden BI-Standard-Lösung nahe. Die Standardisierbarkeit wird durch die Vielfältigkeit der operativen Prozesse, die Heterogenität der Quellsysteme und die Breite der Anwender-Forderungen erschwert und ist vor allem für die BI-Bereiche sinnvoll, die nicht wettbewerbsdifferenzierend wirken.

Automatisierung

Automatisierung als ein weiteres zentrales Feld der Industrialisierung zielt auf die Übernahme von Funktionen und Entscheidungen menschlicher Akteure durch Maschinen. Prozesse sind nur dann sinnvoll zu automatisieren, wenn sie klar definierten Regeln mit einer hohen Wiederholrate ohne menschliche Steuerung folgen. Bezogen auf BI sind sämtliche Generierungs- und Verarbeitungs-Prozesse innerhalb der Informations-Pipeline sowie alle Entwicklungs-, Test-, Deployment-, Dokumentations- und Kontroll-Prozesse auf ihre Automatisierbarkeit zu untersuchen [8].

Im Bereich des Data Warehousing wird meist die Automatisierung der ETL-Strecken an erster Stelle genannt. Da automatisierte Extraktion aus den Quellsystemen, die Transformation und das Beladen in ein Data Warehouse eine hohe Datenqualität voraussetzen, empfehlen neue Ansätze wie Data Vault eine Verlagerung der Transformation in die nachgelagerten Schichten der BI-Infrastruktur [9, Seite 670]. Ganz unvermeidbar ist die Automatisierung dort, wo Prozesse ohne menschliche Einwirkung ablaufen müssen, zum Beispiel im Realtime-DWH.

Neben dem Betrieb der DWH-Infrastrukturen ist die Erstellung der Infrastruktur selbst Gegenstand von Automatisierungsbemühungen. Dafür ist eine Standardisierung der Architekturen und Prozesse sowie ein Metadaten-Management notwendige Basis. Der Gefahr von Performance-Einbußen durch automatisch generierte Strecken stehen Vorteile wie zum Beispiel die Wiederverwertbarkeit, die Einhaltung von Standards und die automatisierte Dokumentationserstellung gegenüber.

Durch die Dominanz von Standard-Systemen im ERP-Bereich können mit der Standard-Software ausgelieferte Extraktoren den Anbindungsprozess erheblich beschleunigen. Eine heterogene, gering standardisierte Quellsystem-Landschaft erschwert eine automatisierte Identifikation und Codierung zu extrahierender Daten. Generische Automatisierungs-Tools erfahren zunehmend Beachtung, sind aber bei den meisten der vom Autor befragten Unternehmen noch nicht im Einsatz. Eine klare Mehrheit der vom Autor befragten Studienteilnehmer sieht in der Automatisierung eine der wichtigsten Aufgaben für BI in den kommenden Jahren.

BI-Organisation

Eine arbeitsteilige Organisation braucht Planung, Koordination und Kontrolle, in der Produktionswirtschaft als „dispositiver Faktor" bezeichnet. Analog besteht für BI weitgehend Konsens darüber, dass eine effektive und effiziente Informationsversorgung im Unternehmen einen organisatorischen Rahmen braucht, aus dem heraus Rollen, Zuständigkeiten beziehungsweise Verantwortlichkeiten für die Formulierung der BI-Strategie und Kontrollmechanismen im Rahmen der Governance definiert werden. Diese bilden die strategischen Vorgaben für Entscheidungen über das Projektportfolio, die Adaption von neuen Technologien und letztlich für alle Fragen zur Industrialisierung wie Sourcing, Make-or-buy, Standardisierung und Automatisierung. Für diese Aufgaben wird häufig eine reale oder virtuelle Organisationseinheit wie das Business Intelligence Competence Center (BICC) postuliert. Die Bedeutung einer effektiven BI-Strategie und BI-Governance wird zunehmend von den Unternehmen erkannt, wie die Ergebnisse der biMA-Studie 2017/18 belegen [2].

Fazit

Die Schwerfälligkeit reaktiv gewachsener BI-System-Strukturen steht den durch Big Data, Analytics und Digitalisierung getriebenen, gegenläufigen BI-Anforderungen gegenüber. Das Bewusstsein wächst, dass BI-Strategien und BI-Governance nötig sind, um die zunehmende Komplexität der BI-Infrastruktur zu beherrschen. BI-Strategie sollte Vorgaben über die Adaptionsbereitschaft neuer Technologien formulieren, eine Sourcing-Strategie beinhalten, darin die Rolle von BIaaS beschreiben und erklären, welche Bereiche zu standardisieren beziehungsweise zu automatisieren sind. Bei der Ableitung dieser Vorgaben dürfte der derzeitige Reifegrad beziehungsweise die strategische Bedeutung der BI-Bereiche für das operative Geschäft eine maßgebliche Rolle spielen.

Nach Zwischenergebnissen der Studie des Autors geht eine deutliche Mehrheit der Befragten davon aus, dass die Bedeutung von BIaaS und Automatisierung der BI-Entwicklung weiter steigen wird. Ebenso wird der BI-Standardisierung und dem Outsourcing eine wachsende Bedeutung zugeordnet, sodass auf eine zukünftige Reduktion der Eigenentwicklung und auf einen vermehrten Einsatz der „Managed Services" geschlossen werden kann. Der damit verbundene Kompetenzverlust in den IT-Abteilungen wird in der Folge zu einem Rückgang deren Relevanz führen. Denn sollte das bekannte, von Antoine de Saint-Exupéry formulierte Muster auf BI zutreffen, dass technologische Entwicklung vom Primitiven über das Komplizierte zum Einfachen erfolgt, so kann man annehmen, dass es irgendwann leichter sein wird, einen BI-Fachanwender mit der nötigen digitale Kompetenz zu versorgen, als einem IT-Spezialisten das nötige fachliche Wissen beizubringen, um den innovativen, wertschöpfenden Einsatz von Informationen in der Geschäftstätigkeit erfolgreich umzusetzen.

Referenzen

[1] Carsten Dittmar, Volker Oßendoth, Klaus-Dieter Schulze, Business Intelligence: Status quo in Europa, europäische biMA-Studie 2012/13; A Steria Report, edited by Steria Mummert Consulting GmbH, 2013: *http://www.bi.soprasteria.de/bi-news-infos/europaeische-bima-studie-2012-13*

[2] Stefan Seyfert, Lars Schlömer, Lisa Anne Schiborr, Zeit für eine neue Kultur durch Business Intelligence & Advanced Analytics, biMA-Studie 2017/18, Edited by Sopra Steria Consulting GmbH, 2017: *https://www.soprasteria.de/newsroom/publikationen/studie/bima-studie-2017-18*

[3] Jens Diekmann, Ursula Besbak, Quo vadis, Data Warehouse? Sanierung statt Neubau als Weg in die Zukunft, In BI-Spektrum 2016 (1), Seiten 30 – 33

[4] BI Trend Monitor 2018, BARC: *http://barc-research.com/wpcontent/uploads/2017/11/BARC-BI_Trend_Monitor_2018-Online.pdf*

[5] Sven Markus Walter, Tilo Böhmann, Helmut Krcmar, Industrialisierung der IT – Grundlagen, Merkmale und Ausprägungen eines Trends, HMD 44 (4), Seiten 6 – 16, 2007.

[6] Ferri Abolhassan, (Hrsg.), Der Weg zur modernen IT-Fabrik, Industrialisierung – Automatisierung – Optimierung, Springer Gabler 2013

[7] Nicole Schirm, Thomas Frank, Manuel Henkel, Frank Bensberg, Erfolgsfaktoren cloudbasierter Business-Intelligence-Lösungen, Wirtschaftsinformatik Proceedings, 2015

[8] Marc Peco, Data Warehouse Automation, Better, Faster, Cheaper ...You Can Have It All. European TDWI Conference, München, 2014

[9] Daniel Linstedt, Michael Olschimke, Building a scalable data warehouse with data vault 2.0, Amsterdam, Morgan Kaufmann, 2016

Hartmut Westenberger
hartmut.westenberger@th-koeln.de

Revisionssichere Optimierung der Datenqualität

Thomas Möller, SüdLeasing GmbH

Die Anforderungen an die Datenqualität haben in den letzten Jahren stark zugenommen. In der Finanzdienstleistungsbranche wird dies sogar konkret von den Aufsichtsbehörden verlangt und überwacht. Auch die Privatwirtschaft ist auf eine hohe Datenqualität angewiesen, da vermehrt Entscheidungen auf Basis von Daten und Algorithmen getroffen werden – Stichwort „Künstliche Intelligenz". Dieser Artikel beschreibt, welchen Weg die SüdLeasing zur systematischen und revisionssicheren Optimierung ihrer Datenqualität eingeschlagen hat.

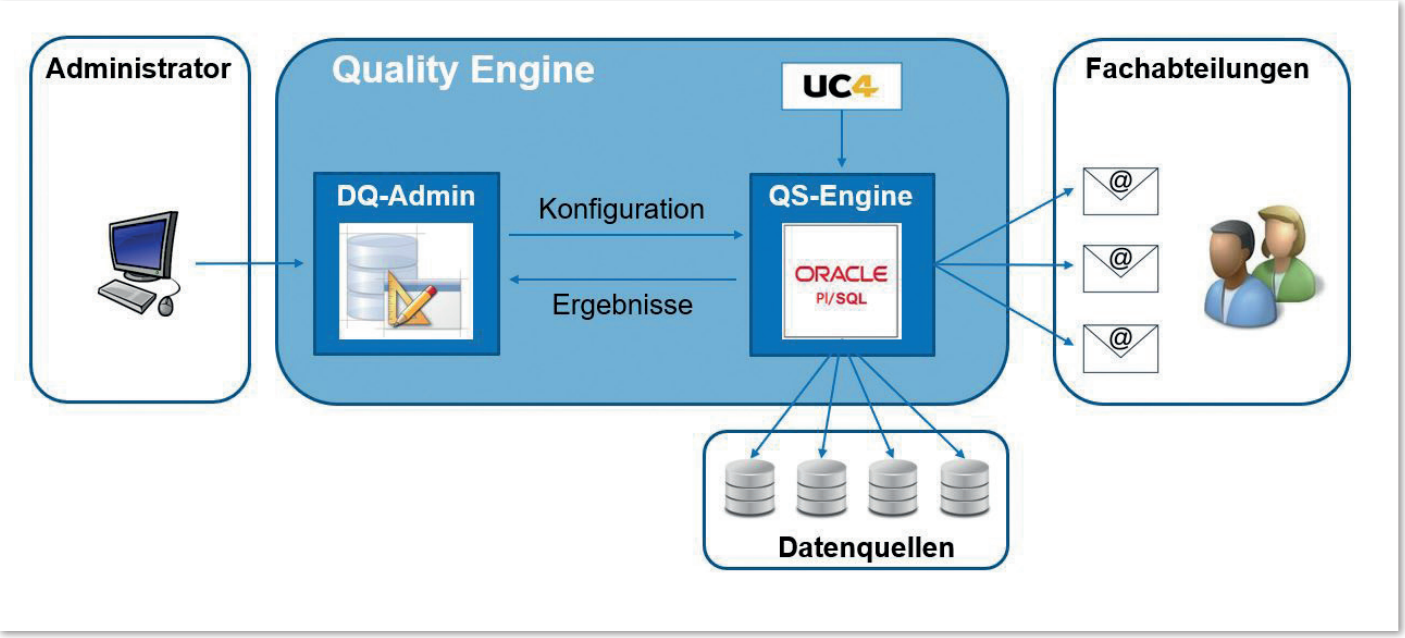


Abbildung 1: Architektonischer Aufbau der Quality Engine

Im Jahr 2013 fand im Unternehmen eine Bestandsaufnahme rund um das Thema „Datenqualität" statt. Dabei hat man eine Vielzahl bestehender Daten-Selektionen gefunden. So wurden neben diversen parametergesteuerten Berichten über den Oracle Discoverer zusätzlich einige Apex-Anwendungen identifiziert, mit denen Mitarbeiter parametergesteuert Daten abfragen konnten. Darüber hinaus wurden Excel-Listen sowie eine Access-Datenbank zur Daten-Sammlung und -Analyse verwendet. Ergänzt wurde das Ganze durch externe Berichte, die von der Landesbank Baden-Württemberg (Konzernmutter) bereitgestellt wurden. Zur Dokumentation diente ein mehr als zweihundert Seiten umfassendes Word-Dokument.

Diese Bestandsaufnahme machte deutlich, dass zwar eine Vielzahl von Daten-Auswertungen durchgeführt wurden, es jedoch nicht wirklich nachvollziehbar war, wer wann welche Abfrage mit welchen Parametern ausgeführt und dabei welche Ergebnisse erhalten hat. Durch die häufige Verwendung von „SYSDATE" war später auch nur über Anpassung des SQL-Statements nachvollziehbar, welche Ergebnisse wahrscheinlich zu einem früheren Zeitpunkt ermittelt wurden. Der laufende Aufwand zur Pflege der Dokumentation war ebenfalls nicht unerheblich; trotzdem hinkte diese normalerweise der tatsächlichen Implementierung hinterher.

Grundprinzipien

Auf Basis der Kritikpunkte wurde beschlossen, das Qualitätsmanagement auf eine neue Basis zu stellen. Dazu wurden folgende Kernforderungen vorgegeben:

- *Keine Notwendigkeit zur Einbindung der IT bei Anpassung der Prüfregeln*
  Die wesentlichste Anforderung: Anpassungen an neue Erkenntnisse müssen zeitnah außerhalb teilweise langwieriger Prozesse erfolgen können. Schnelle Reaktionen auf neue Erkenntnisse sind für die Akzeptanz des Systems bei den Endanwendern essenziell.

- *Hohe Konfigurierbarkeit durch Endanwender*
  Dieser Punkt zielt in dieselbe Richtung: Die Anpassung von beispielsweise Wertelisten oder Benutzer-Berechtigungen muss ohne ein Release erfolgen können. Nur so ist eine agile Vorgehensweise möglich.

- *Sicherstellung der Revisionssicherheit*
  Das neue System soll lückenlos belegen können, wer wann welche Änderung vorgenommen hat. Außerdem wurde das Vier-Augen-Prinzip bei Produktivsetzung neuer oder geänderter Prüfregeln gefordert. Darüber hinaus sollen die Ergebnisse der Prüfregeln ebenfalls dauerhaft gespeichert sein.

- *Automatische Erstellung der Dokumentation*
  Ziel ist, alle Informationen so in einer Datenbank zu speichern, dass jederzeit (auf Knopfdruck) der aktuelle Stand aller Qualitätsprüfungen als Dokumentation bereitgestellt werden kann.

- *Modularer Aufbau*
  Das System soll so aufgebaut sein, dass jederzeit weitere Bausteine hinzugefügt und/oder diese neu zusammensetzt werden können.

Architektonische Umsetzung

Die Quality Engine besteht aus zwei Komponenten: Die Apex-Anwendung „DQ-Admin" stellt die Verwaltungs-Komponente dar und verwaltet Prüfregeln, Benutzer, Ausnahmeregeln etc.; die Komponente „QS-Engine" besteht aus einer Sammlung von Packages und Prozeduren in PL/SQL und führt die aktuellen Prüfregeln aus *(siehe Abbildung 1)*.

Änderungen und Erweiterungen von Prüfregeln werden zentral vom Administrator über den DQ-Admin verwaltet. Zu den vorgegebenen Zeiten (meistens nachts) startet ein Scheduler (hier „UC4") die Ausführung der QS-Engine. Diese ruft im ersten Schritt die aktuellen Informationen aus dem DQ-Admin ab. Anschließend werden alle Prüfregeln in Form von SQL-Statements gegen die definierten Datenquellen ausgeführt. Alle ermittelten Ergebnisse werden dabei abgespeichert und dem DQ-Admin für weitere Analysen zur Verfügung gestellt. Zuletzt prüft die QS-Engine, ob und zu welchen Prüfregeln Fachbereiche informiert werden müssen, und stellt diesen gefundene Regelverletzungen zusammen mit erläuternden Informationen per E-Mail bereit.

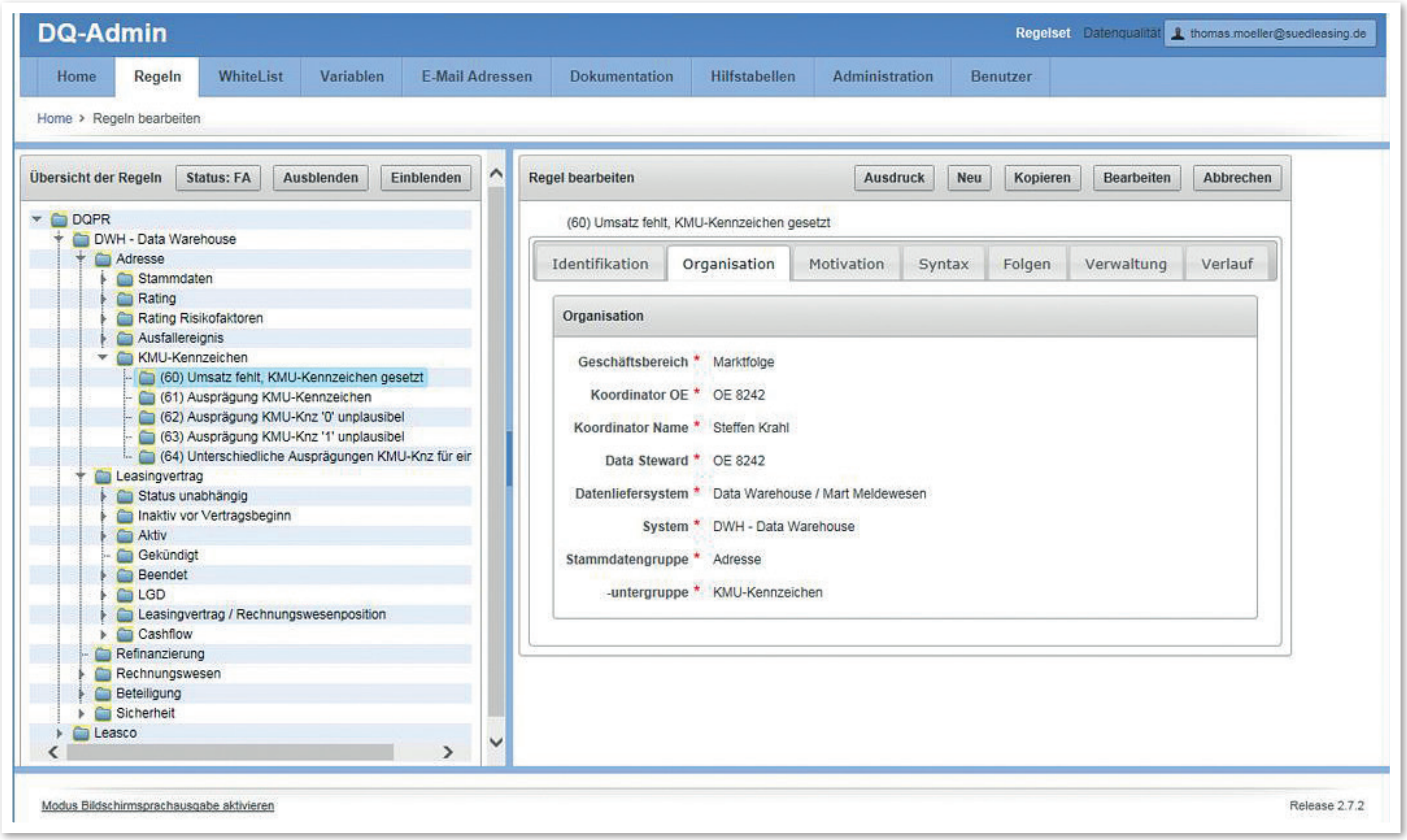


Abbildung 2: Pflege von Prüfregeln im DQ-Admin

Der DQ-Admin – die Steuerzentrale

Die Anwendung „DQ-Admin" verwaltet die Prüfregeln. *Abbildung 2* zeigt auf der linken Bildschirmseite die Auflistung der Prüfregeln in einer Baumstruktur. Diese Anordnung hat sich beim Suchen nach einzelnen Prüfregeln als hilfreich erwiesen. Auf der rechten Seite sind die Felder zur Pflege der einzelnen Informationen zu einer Prüfregel auf Registerkarten angeordnet. Die Vielzahl der Informationen, die eine Prüfregel beschreibt, machte diese Anordnung notwendig.

Jede Prüfregel erhält eine eindeutige Nummer. Diese ist im Gegensatz zur Bezeichnung der Prüfregel unveränderlich. Der Name einer Prüfregel soll den Sachverhalt klar benennen, den diese Regel überprüft. Eine Regel heißt also beispielsweise nicht „Umsatz", sondern klar „Mindestumsatz unterschritten". Auf diese Weise werden Interpretations-Spielräume reduziert und Irrtümer über Inhalt und Aufgabe der Prüfregel ausgeschlossen. Über das „Aktiv-Flag" einer Regel kann diese kurzfristig deaktiviert oder auch eine vorbereitete Regel einfach produktiv genommen werden.

Zur einfachen Prüfung daraufhin, welche Regeln dem Vertrieb, der Marktfolge oder den Stäben zugewiesen sind, sind Prüfregeln bei der SüdLeasing jeweils einem Geschäftsbereich zugeordnet. Der Koordinator einer Prüfregel identifiziert dabei diejenige Person im Qualitäts-Management, die für die Pflege der Regel verantwortlich ist. Spannend ist die Auswahl des Data-Stewards. Dieser ist Empfänger der Ergebnisse der Prüfregel und für die Korrektur der fehlerhaften Daten verantwortlich. Grundsätzlich wird nur die Organisationseinheit gespeichert. An anderer Stelle werden der Organisationseinheit die konkreten Personen mit ihrer E-Mail-Adresse zugewiesen. Als Empfänger werden nur Führungskräfte akzeptiert, also mindestens Gruppenleiter. Die Führungskräfte müssen die Korrektur der Daten nicht selbst vornehmen, sie tragen allerdings die Verantwortung dafür.

Ein weiteres wichtiges Feld ist das „Datenliefersystem". Es steuert, in welcher Datenbank die Abfrage ausgeführt wird. Die Quality Engine kann grundsätzlich auf alle Datenquellen in der SüdLeasing zugreifen. Dazu ist in jeder Datenbank ein Package mit dem Namen „QS-Agent" installiert. Es führt die Regeln lokal in der Datenbank aus und liefert die gefundenen Treffer zurück an die QS-Engine.

Die nächsten Informationen, die bei der Prüfregel gespeichert werden, sorgen für die Einordnung der Regel in die dreistufige Baumstruktur. Die Auswahl des Prüfobjekts erlaubt es, später auszuwerten, welche Prüfungen zu einem Datenfeld implementiert sind. Über das Prüfverfahren wird gesteuert, ob die Regel durch die QS-Engine ausgeführt wird. Auf diesem Weg können auch Prüfmaßnahmen, die außerhalb der Quality Engine implementiert sind, zentral an einer Stelle dokumentiert werden.

Der nächste wichtige Punkt heißt „Hintergrund". Es bietet sich an, hier ausführlich den Sachverhalt zu beschreiben. Dieser Text wird als E-Mail an den Data-Steward ausgegeben. Im Feld „Resultat" wird das erwartete Ergebnis der Prüfregel dokumentiert. Für Prüfregeln, die gezielt Fehler suchen, ist hier zu lesen, dass keine Datensätze erwartet werden. Es gibt aber auch Prüfregeln vom Typ „Sichtprüfung", deren Ergebnis ein Sachbearbeiter visuell prüft.

Als Nächstes geht es darum, die Regel technisch zu beschreiben. Hierzu wird im Feld „Syntax" die Regel in Pseudocode beschrieben. Dies erleichtert es zu verstehen, welche Prüfung die Regel konkret vornimmt. Im Feld „Kriterien" sind die Bedingungen aufgelistet. Dieses Feld verbindet den Pseudo-Code mit dem SQL.

In weiteren Feldern wird das eigentliche SQL-Statement abgelegt. Aus technischen

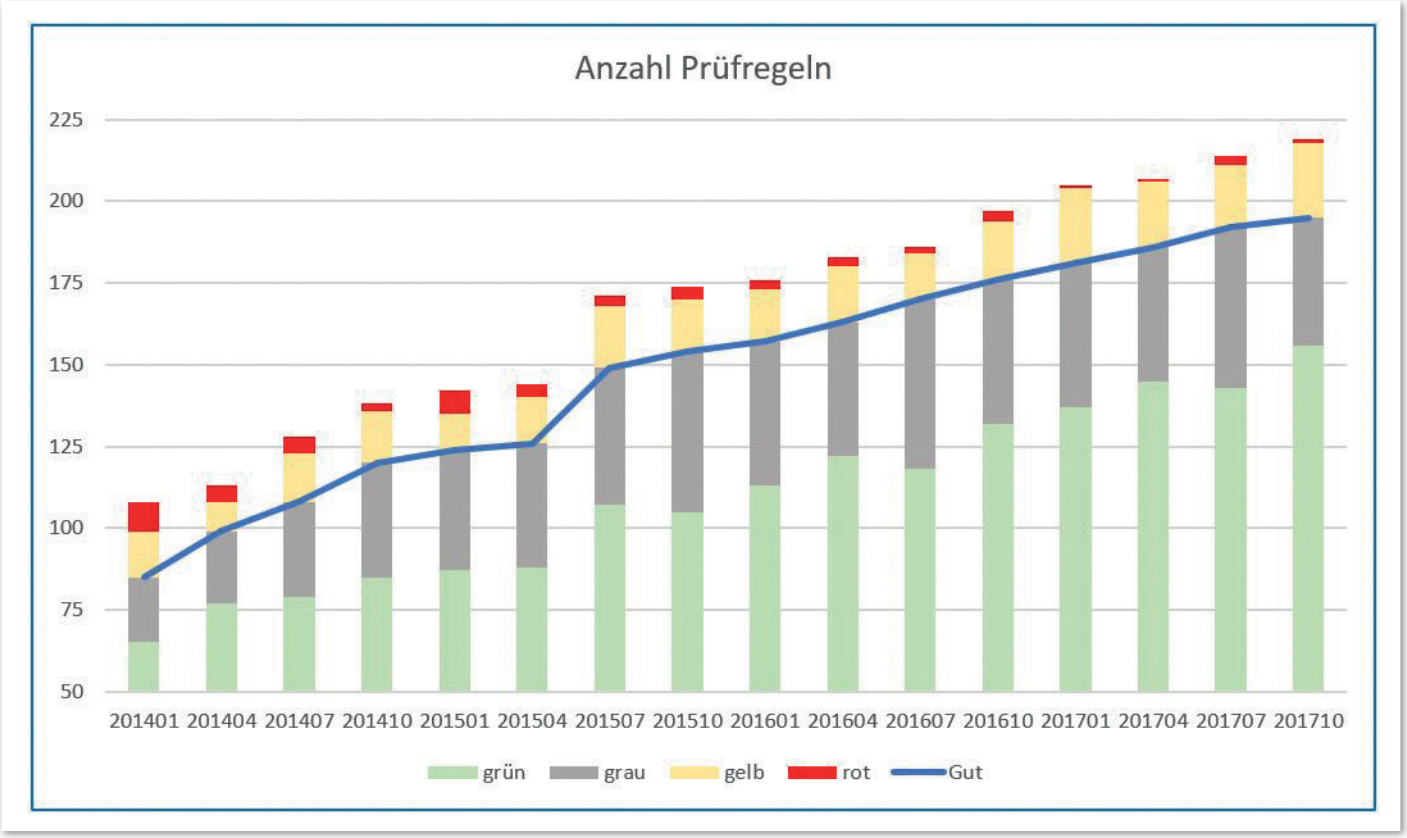


Abbildung 3: Entwicklung der Prüfregeln im Zeitverlauf

Gründen wird das Statement in seine einzelnen Bestandteile („Select", „From", „Where" etc.) zerlegt. Ein kleiner Assistent hilft bei dieser Arbeit. Ein fertiges SQL-Statement, das etwa in der IT entwickelt wurde, kann so auf einfachem Weg von einem Sachbearbeiter in seine Einzelteile zerlegt werden.

Das SQL-Statement kann Variablen enthalten, die zum Zeitpunkt der Ausführung durch ihren aktuellen Wert ersetzt werden. Neben den Systemvariablen wie dem Tagesdatum können auch eigene Variablen definiert werden. Im Feld „Spaltenüberschrift" sind die Bezeichnungen der Datenfelder als kommaseparierte Liste gespeichert. Dieses Feld wird in der „.csv"-Datei als Überschriftenspalte verwendet. Das Feld „Identifier" bezeichnet das Feld, das einen gefundenen Treffer eineindeutig identifiziert. Auf diesem Weg kann später ermittelt werden, seit wann ein Treffer von der QS-Engine gefunden wird und ob der gleiche Fall bei mehreren Prüfregeln auftritt.

Im Datenfeld „Meldung" wird der Text erfasst, der später dem Data-Steward als Handlungsanweisung übermittelt wird. Im Feld „Auswirkungen" sind die Auswirkungen eines fehlerhaften Datensatzes grundsätzlich kategorisiert. Über den „Schweregrad" und die „Reaktionszeit" wird deutlich, wie wichtig die Bereinigung des Fehlers ist. Der „Versandrhythmus" legt fest, wie häufig die Ergebnisse an den Data-Steward übersandt werden.

Zu jeder Prüfregel wird automatisch gespeichert, wann und durch wen diese angelegt wurde. Auch wann und durch wen die letzte Änderung erfolgt ist, wird festgehalten. Das Datum und der Mitarbeiter der letzten Freigabe runden die automatisch gespeicherten Informationen zu einer Prüfregel ab. Das Feld „Änderungsgrund" gibt einen Hinweis darauf, welche Änderungen gegenüber der letzten Version der Regel vorgenommen wurden.

Eine Version einer Prüfregel wird nur dann produktiv genommen, wenn diese freigegeben wurde. Die Freigabe erfolgt nach dem Vier-Augen-Prinzip. So ist sichergestellt, dass jede Änderung einer Qualitätssicherung unterliegt. Über die Oberfläche des DQ-Admin kann die Konfiguration geändert werden. Viele der genannten Datenfelder sind als Dropdown-Felder implementiert. Die Benutzer sind in der Lage, die jeweiligen Wertelisten zu ergänzen und bestehende Begriffe anzupassen.

Erste Erfahrungen

Bei der Arbeit mit der Quality Engine wurde schnell deutlich, dass es eine Vielzahl von Bereichen gibt, für die die Quality Engine zum Einsatz kommen kann. Um zu vermeiden, dass die verschiedenen Interessengruppen gegenseitig die Prüfregeln ändern oder die Einträge der Dropdown-Felder anpassen, wurde die Anwendung mandantenfähig ausgestaltet. Ein Mandant nennt sich „Regelset". Das ist ein bestandstrennendes Merkmal. Beim Öffnen der Anwendung wird das Regelset abgefragt.

Die Praxis hat gezeigt, dass es Sachverhalte gibt, die zwar gemäß der Definition der Prüfregel falsch sind, die sich jedoch nachträglich nicht mehr ändern lassen. Um zu verhindern, dass diese Sachverhalte wieder und wieder in den Ergebnislisten für den Fachbereich auftauchen, wurde das Instrument der „WhiteList" geschaffen. Die Einträge darin unterliegen dem Vier-Augen-Prinzip. Sachverhalte, die auf der WhiteList stehen, werden zwar von der QS-Engine gefunden, jedoch nicht an den Data-Steward gemeldet.

Über eine Benutzerverwaltung ist ein Rechtesystem abgebildet. So kann jedem Benutzer eine Rolle zugewiesen werden, die es ihm erlaubt, ein Regelset „read-only" zu öffnen, Regeln zu bearbeiten oder gar freizugeben und gegebenenfalls als Regelset-Administrator sogar die Einträge in der Konfiguration anzupassen.

Der DQ-Admin erlaubt es, die gefundenen Ergebnisse zu analysieren. Hierzu kann der Benutzer in einem Report beliebige Filter setzen, Sortierungen vornehmen und die so eingeschränkten Daten zum Beispiel nach Excel exportieren.

Der DQ-Admin stellt auf Knopfdruck eine Dokumentation bereit. Hierzu werden alle gespeicherten Informationen über den BI-Publisher als PDF-Dokument ausgegeben. Da für die Dokumentation dieselben Informationen wie für die Ausführung der Prüfregeln verwendet werden, ist die Dokumentation automatisch immer auf dem aktuellen Stand.

Die QS-Engine – die Arbeitsbiene

Die QS-Engine ist der Motor der Quality Engine. Sie sorgt dafür, dass die Prüfregeln jede Nacht ausgeführt und die Ergebnisse den Fachbereichen zur Verfügung gestellt werden. Im ersten Schritt werden dazu die Daten aus dem DQ-Admin geladen. Danach erfolgt die Prüfung der geladenen Daten auf Konsistenz. Im nächsten Schritt werden alle Prüfregeln ausgeführt und die Ergebnisse gesichert. Im Nachgang wird für jede Prüfregel geprüft, ob der Fachbereich informiert werden muss. Dazu ist für jede Regel ein Versandrhythmus definiert. Als Rhythmen stehen „täglich", „wöchentlich" und „monatlich" zur Verfügung. Die Auswahl erfolgt je nach Kritikalität.

Sobald der definierte Termin erreicht ist, wird eine E-Mail an den hinterlegten Data-Steward versendet. Darin wird der Hintergrund der Prüfregeln erläutert. Abgeschlossen wird die E-Mail mit einer konkreten Handlungsaufforderung. Die gefundenen Treffer sind als „.csv"-Datei angehängt. Ein Doppelklick darauf öffnet die Datei in Excel und zeigt an, welche Daten korrigiert werden müssen.

Der Administrator eines Regelsets erhält zusätzlich eine Übersichts-Mail. Darin sind die Ergebnisse aller Prüfregeln einschließlich des Vortagesstands aufgelistet. Eine zusätzliche Spalte markiert größere Abweichungen. Mit diesem Instrument hat der Administrator schnell einen aktuellen Überblick über den Gesamtbestand eines Regelsets.

Erkenntnisse und Erfahrungen

Die Quality Engine ist mittlerweile seit mehr als vier Jahren im Einsatz. Während dieser Zeit wurden folgende Erkenntnisse gewonnen:

- *Mach es schriftlich*
  Es ist wichtig, viele Details rund um eine Prüfregel zu dokumentieren. Im Moment der Definition einer Regel sind alle Details für die Beteiligten klar. Wenn es aber in einigen Monaten eine Nachfrage gibt oder die Regel angepasst werden soll, sind diese Informationen sehr hilfreich.

- *Mach es jede Nacht*
  In der SüdLeasing werden alle Prüfregeln jede Nacht ausgeführt. So lässt sich der Verlauf einer Prüfregel detailliert nachvollziehen. Außerdem gibt es Konstellationen, die nur selten auftreten. Durch die regelmäßige Ausführung der Prüfregeln ist sichergestellt, dass auch diese erfasst werden.

- *Finde den Data-Steward*
  Eine der schwierigsten Aufgaben bei der Definition einer Prüfregel ist es, den Data-Steward zu finden. Es besteht häufig Einigkeit darüber, dass ein Sachverhalt korrigiert werden muss. Diese Einigkeit ist allerdings schnell erschöpft, wenn es darum geht, wer die fehlerhaften Daten korrigieren soll. Hier ist eine gute Data-Governance hilfreich.

- *Erkenne die Muster*
  Über die grafische Abbildung, an welchen Tagen es wie viele Treffer in einer Prüfregel gegeben hat, lassen sich Muster erkennen. Ideal ist ein kontinuierlich sinkender Verlauf. Typisch ist auch ein sägezahnartiger Verlauf. Dieser deutet darauf hin, dass an einem bestimmten Tag im Quartal, im Monat oder in der Woche etwas passiert, das zu einer Vielzahl von Treffern führt, die im nachfolgenden Zeitraum korrigiert werden. Auch die seltenen Treffer, die nur ab und an auftauchen, werden über eine grafische Abbildung schnell sichtbar.

- *Incidents sind Trigger für neue Regeln*
  Es ist sinnvoll, das Qualitätsmanagement in den Incident-Prozess einzubinden. Wenn ein Incident korrigiert oder dafür vielleicht sogar ein Hotfix gemacht wird, ist dies ein Zeichen dafür, dass ein kritischer Prozess fehlerhaft war. Warum also nicht prüfen, ob sich der korrigierte Zustand mithilfe einer Prüfregel überwachen lässt? Sollte der Zustand erneut auftauchen, so wird dies durch die Prüfregel sofort bemerkt. Auf diesem Weg gewinnt man dauerhaft die Sicherheit, dass ein erkannter und korrigierter Fehler nicht unbemerkt zurückkehrt.

Erfolge

Es stellt sich die Frage nach dem Nutzen. Dieser wird in *Abbildung 3* verdeutlicht. Für alle Prüfregeln wird jeweils eine Fehlerquote ermittelt. Dazu wird die Anzahl der fehlerhaften Datensätze der jeweiligen Basismenge gegenübergestellt. Die Fehlerquoten sind farblich kodiert. Der Bereich zwischen 1 und 5 Prozent wird in der Grafik gelb dargestellt. Fehlerquoten größer als 5 Prozent sind rot, Regeln mit 0 Treffern grün kodiert.

Regeln mit mehr als einem und weniger als 1 Prozent Fehler sind grau dargestellt. Diese Grauzone ist absichtlich gewählt: Knapp 1 Prozent Fehler, bezogen auf eine Basismenge von 11.000 Datensätzen, bedeutet, dass es mehr als 100 fehlerhafte Datensätze gibt. Je nach Sachverhalt kann diese Anzahl ein nicht unerhebliches Konfliktpotenzial mit sich bringen. Die Farbe Grau soll verhindern, dass Führungskräfte bei einem flüchtigen Blick auf die Statistik dem Eindruck erliegen, alles sei im grünen Bereich.

Die *Abbildung 3* zeigt, dass die Anzahl der Prüfregeln stetig zunimmt. Gleichzeitig geht die Anzahl der gefundenen Fehler sukzessive zurück. Die Data-Stewards sind entspannt. Sie wissen genau um die noch offenen Problembereiche. Hektische Bereinigungsaktionen hat es lange nicht mehr gegeben. Auch die Zufriedenheit der Steakholder ist gestiegen. Zu guter Letzt hat sich die Außenwahrnehmung im Bereich „Datenqualität" gebessert.

Fazit

Die Sicherstellung der Datenqualität ist eine Daueraufgabe. Nur wer sich darauf einlässt und seine Daten systematisch und nachhaltig bereinigt, wird am Ende erfolgreich sein. Die SüdLeasing hat mit der Quality Engine ein Tool geschaffen, das diese Aufgabe perfekt unterstützt. Die erreichten Erfolge zeigen, dass dieser Weg richtig ist. Die gewonnenen Erkenntnisse und Erfahrungen dürfen gern auf die eigene Situation angewendet werden.

Thomas Möller
thomas.moeller@suedleasing.com

„noETL, yesSQL“ – warum ELT und SQL die optimale Wahl für ein modernes Data Warehouse sind

Alec Shalashou, datasqill

Dieser Artikel beschäftigt sich mit der Analyse unter Verwendung des ELT-Verfahrens (extract, load, transform) und des Einsatzes von SQL als Transformationssprache im Kontext moderner DWHs. Der Artikel stellt das ELT-Verfahren dem gängigen ETL-Verfahren (extract, transform, load) gegenüber und vergleicht die Transformationsentwicklung mit SQL mit der grafischen Transformationsentwicklung. Er stellt Beispiele für ELT-Transformationslandschaften vor und macht Vorschläge für die sinnvolle Umsetzung des ELT-Verfahrens sowie der Entwicklung mit SQL in einem Data Warehouse (DWH).

Die Welt des modernen Data Warehousing ist dynamisch wie noch nie zuvor. Die Datenmengen steigen und die Komplexität der analytischen Fragestellungen wächst kontinuierlich weiter an. Selbst kleine und mittelständische Unternehmen stehen inzwischen vor anspruchsvollen analytischen Anforderungen.

Viele Unternehmen setzen heute auf agile Entwicklungsmethoden, um die schneller wechselnden und wachsenden Kunden-Anforderungen bedienen zu können. Dies verkürzt Entwicklungs- und Release-Zyklen, verringert Risiken durch Prototyping der Lösungsansätze und stellt Ergebnisse wie Reports und Analysen für Endanwender schneller bereit. Das Ganze wird unterstützt durch die kontinuierliche Verbesserung und Spezialisierung der verfügbaren Werkzeuge für die verschiedenen Einsatzgebiete in DWH-Entwicklungsprojekten.

In solch einem dynamischen und datenintensiven Umfeld sieht sich ein Entwicklungsteam häufig mit der Frage konfrontiert, mit welchen Ansätzen und Tools sich der Datenaufbereitungs-Prozess in einem DWH am effizientesten gestalten lässt. Eine mögliche Antwort sind die Konzepte, die sich unter dem von uns vorgeschlagenen Begriff „noETL, yesSQL“ verbergen:

- *noETL*
  Ausführung von Datentransformationen dort, wo die Daten gespeichert sind
- *yesSQL*
  Programmierung der Transformationslogik in SQL

noETL

Mit dem Aufkommen der Data-Warehouse-Lösungen kommen ETL-Prozesse für ihre Befüllung und Bewirtschaftung zum Einsatz. Dabei überträgt ein Integrationsserver die Daten aus den Quellsystemen in die Datenbank des DWH. Da der Datenspeicher begrenzt und teuer ist, werden die Daten gleich bereinigt, transformiert und in geeigneter Form gespeichert *(siehe Abbildung 1)*.

Mit der Zeit wuchsen die Datenbestände in Data-Warehouse-Systemen immer weiter an. Zur logischen Strukturierung der Daten entstanden die heute verbreiteten Schichten für Staging, 3NF-Layer und Data Marts. Die Entwicklung und Ausführung der Transformationen erfolgt nach wie vor mit den etablierten ETL-Werkzeugen. Dabei kommt folgendes Verfahren zum Einsatz *(siehe Abbildung 2)*:

- Die Daten werden aus den Quellsystemen extrahiert („extract“)
- Die Daten werden auf dem Application-Server des ETL-Tools transformiert („transform“)
- Die Daten werden in das Zielsystem geschrieben („load“)

Bei dem beschriebenen Verfahren ist die DWH-Datenbank sowohl Quell- als auch Ziel-System für die Mehrzahl der Transformationen. Gerade bei einer mehrstufigen Verarbeitungslogik kann das zum wiederholten Lesen von Daten aus dem DWH in den Application-Server führen, die in vorherigen Schritten von dort in die Datenbank geschrieben wurden.

Viele ETL-Tools bieten eine grafische Transformationssprache, die allerdings proprietär ist und oft mit weiteren proprietären Programmiersprachen kombiniert wird. Der ausgeführte Transformationscode ist für den Entwickler nicht immer transparent und nachvollziehbar. Einige Tools generieren Code für Standard-Programmiersprachen wie zum Beispiel Java. Um die übersetzte Transformationslogik nachzuvollziehen und im Fehlerfalle analysieren zu können, müssen viele Datenbank-Entwickler daher zunächst eine neue Technologie verstehen und sicher beherrschen lernen.

Wie bewährt sich die Datentransformation mit ETL unter modernen Aspekten? Zunächst einmal arbeiten viele ETL-Tools zeilenorientiert, was im Zusammenspiel mit dem wiederholten Lesen und Schreiben von Daten zu entsprechenden Einbußen bei der Performance führt und sich spätestens bei der Verarbeitung von Massendaten als unüberwindbarer Hemmschuh erweist. Zweitens ist der Lösungsansatz, bei dem die Daten aus dem Datenspeicher ausgelesen werden, oft schwerfällig oder ungeeignet für neue Technologien wie Cloud-, In-Memory-, MPP-Datenbanken oder Big Data. Zum Dritten werden die vorhandenen Ressourcen und das Potenzial des Datenbank-Servers nicht optimal genutzt. Viertens entsteht durch die proprietäre Transformationsumgebung und aufgrund fehlender Standardisierungen ein starkes Vendor-Lock-in zu oft hochpreisigen ETL-Produkten. Die Entwickler müssen neben SQL auch die grafische Sprache des ETL-Tools beherrschen, was den Entwicklungsprozess ebenfalls verteuert und die Suche nach geeigneten Mitarbeitern erschwert.

Als Alternative zu ETL bietet sich ELT als Datenverarbeitungsverfahren für Data Warehousing an. Dabei werden die Daten aus den Quellsystemen extrahiert und ohne Transformation ins Data Warehouse geladen. Dort finden alle weiteren Transformationsschritte im Datenspeicher statt *(siehe Abbildung 3)*.

Das Performance-Verhalten bei ELT ist für Massendaten besser ausgelegt, da die komplexe Datenverarbeitung dort stattfindet, wo die Daten gespeichert sind. Es vermeidet Netzwerk-Verkehr und die bereits beschriebenen Schreib-/Lese-Vorgänge bei Transformationen – was gerade für Cloud-Lösungen, Massendaten-Verarbeitung und Big-Data-Lösungen für entsprechende Performance sorgt und ideale Voraussetzungen zur Skalierung mit sich bringt. Auch bei Architekturen mit In-Memory- und MPP-Datenbanken nutzt ELT die Features und Optionen der Datenspeicher optimaler aus, was zugleich zum Schutz der getätigten Investitionen in die Datenbank-Technologie beiträgt. Eine für Datenspeicher native Transformationssprache (wie SQL bei Datenbanken) erlaubt die einfachere Entwicklung und Fehler-Analyse, da der Quellcode der Transformationen direkt in der Zielumgebung ausführbar ist. Das minimiert zudem das Vendor-Lock-in, da der Transformationscode standardisiert und portabel ist.

Moderne Datenbanken stellen gute ELT-Plattformen dar, sie sind ausgelegt für die Verarbeitung von Massendaten und bieten Algorithmen zur Optimierung der Aus-

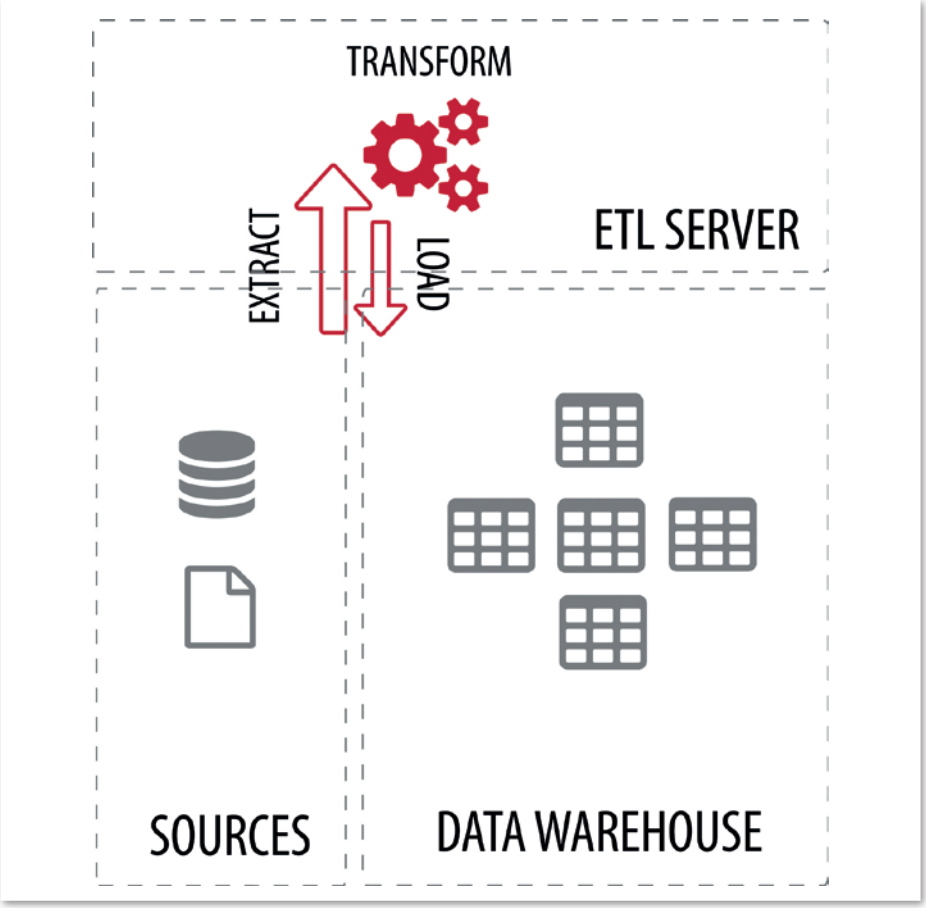


Abbildung 1: Erster DWH-Ansatz

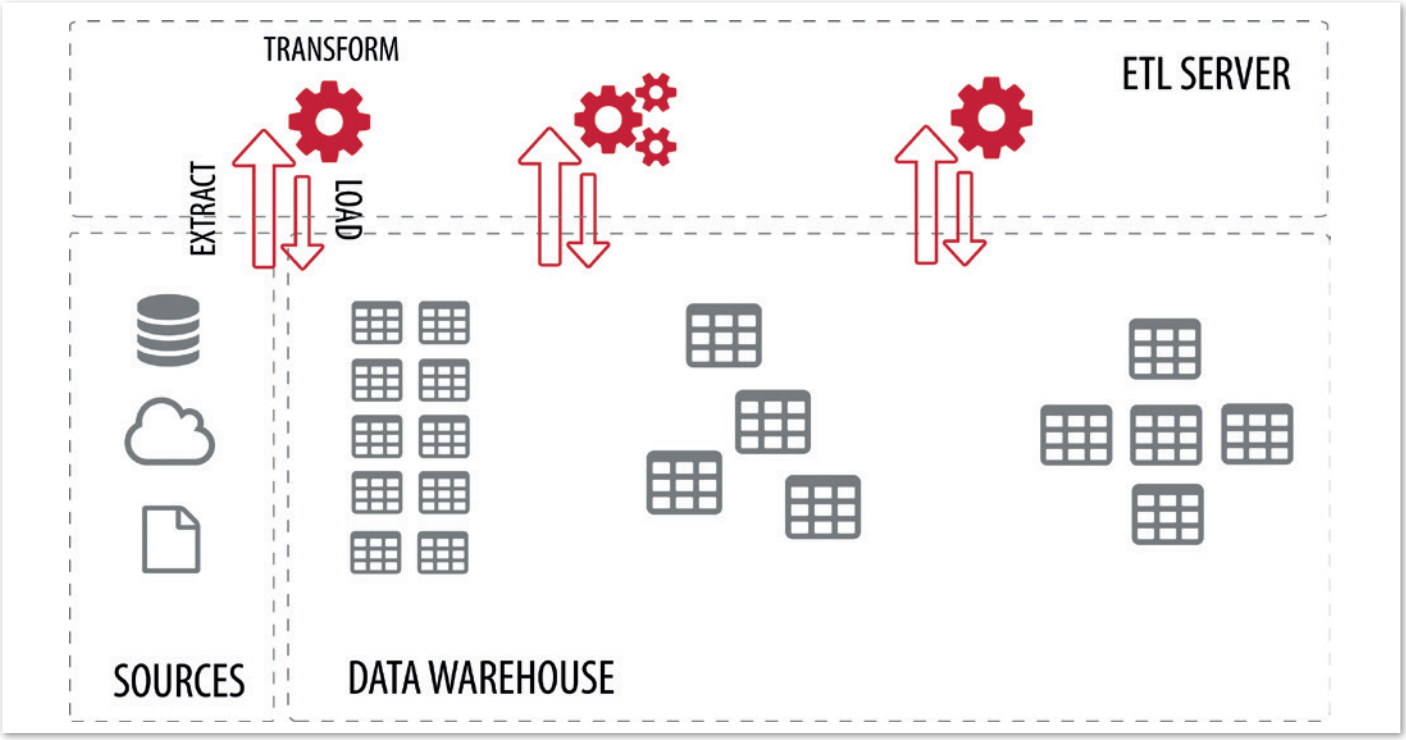


Abbildung 2: ETL in einem mehrschichtigen Data Warehouse

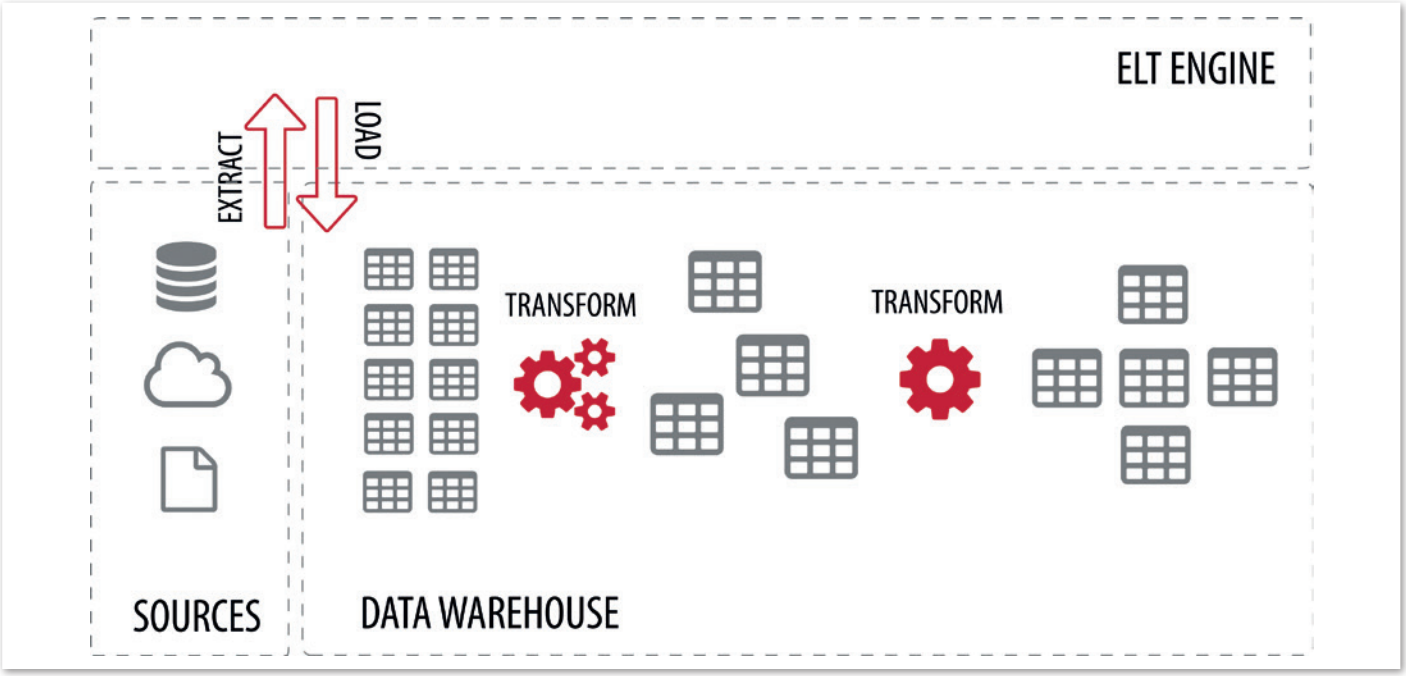


Abbildung 3: ELT in einem mehrschichtigen Data Warehouse

führungspläne für größere Datenmengen. Speicherplatz ist heutzutage keine teure Ressource mehr. Column-based-, In-Memory- und MPP-Datenbanken sind speziell für Data Warehousing und Daten-Analysen ausgelegt. Viele Datenbanken bieten inzwischen Integrationskomponenten wie Bulk-Loader und CSV-, XML- oder JSON-Parser.

Datenbanken können in der Cloud betrieben werden und skalieren dort nach Bedarf. Dadurch haben sich neue, auf Data Warehousing spezialisierte Cloud-Datenbanken (wie Autonomous Data Warehouse von Oracle, Amazon Redshift, Snowflake) auf dem Markt etabliert. Die Entwicklung der Datenbank-Technologie ist also sehr lebendig und noch lange nicht abgeschlossen, die bestehenden Datenbanken werden kontinuierlich weiterentwickelt, daneben kommen immer wieder neue Technologien und Produkte hinzu.

yesSQL

Wenn man die Datenbank als ELT-Plattform betrachtet, bietet sich SQL als Transformationssprache ganz natürlich an. Es ist eine verbreitete Sprache, die mächtige Transformationsmöglichkeiten (wie analytische Funktionen) mitbringt. ANSI SQL ist als Standard etabliert, was SQL zu einer universellen Abfragesprache für heterogene Datenbanken macht. Nicht nur Datenbanken

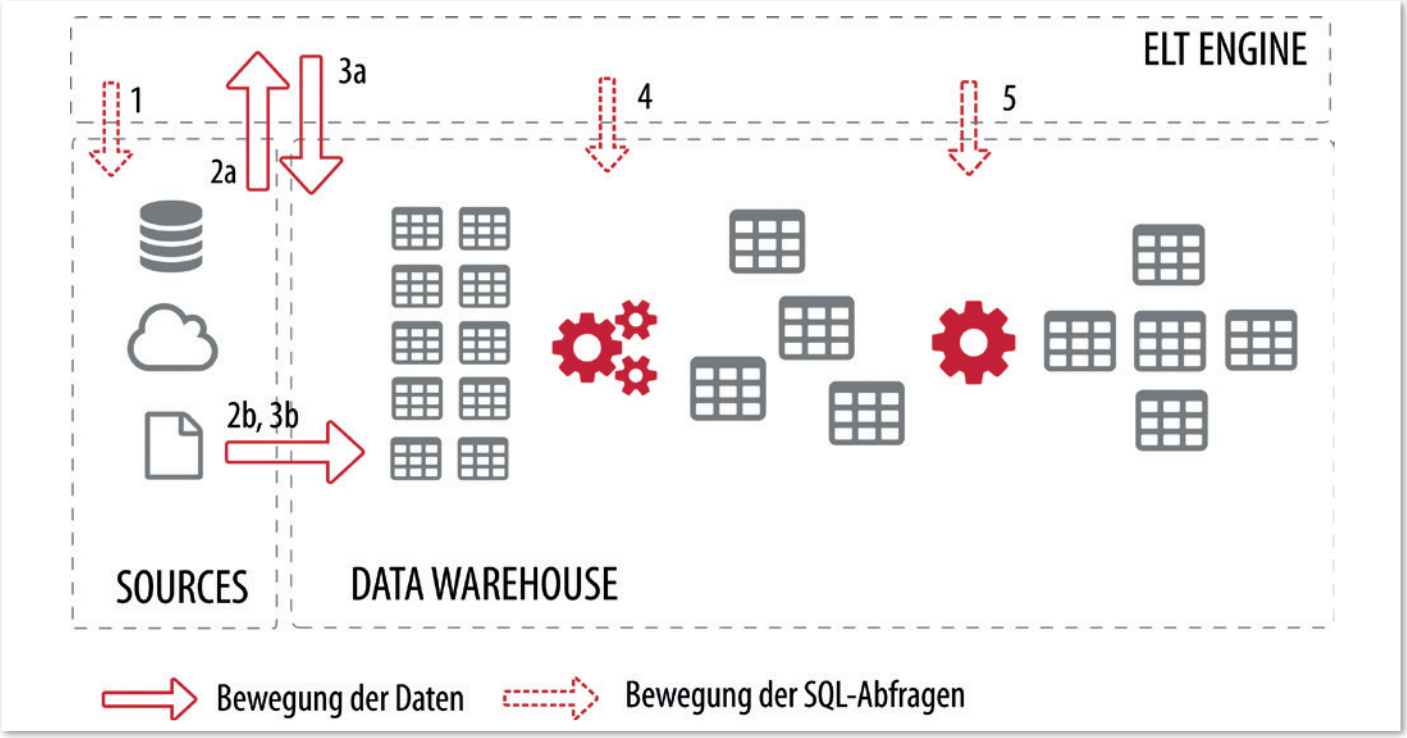


Abbildung 4: „noETL, yesSQL" im Data Warehouse

verstehen SQL, es gibt viele Big-Data-SQL-Tools (Oracle Big Data SQL, Hive, Impala, Spark SQL), SQL-basierte Flat-File-Integrationstools (Apache Drill) und die Mehrzahl der BI-Tools unterstützt die Verwendung von SQL.

SQL ist eine sehr verbreitete Sprache, was für ein größeres Angebot an Mitarbeitern mit SQL-Kenntnissen im Gegensatz zu proprietärem ETL-Tool-Know-how sorgt.

Die Transformationsentwicklung mit SQL verläuft schneller und agiler als die Entwicklung mit grafischen Tools, da die Analyse, der Prototyp und die Implementierung in der gleichen Sprache erfolgen. So entsteht kein Medienbruch zwischen dem während der Analyse entwickelten Code (SQL) und dem produktiv laufenden Code (proprietärer Code bei Verwendung eines ETL-Tools). Der Transformation-Code in SQL ist portabel und in der Regel kompakter und übersichtlicher als grafische Mappings. Damit eignet er sich auch besser für komplexere Logik.

„noETL, yesSQL" im Data-Warehousing-Kontext

Data-Warehouse-Lösungen bieten einen perfekten Rahmen für das „noETL, yesSQL"-Konzept, weil die Daten in einer oder wenigen Datenbanken gespeichert sind. Solche Lösungen sind Massendaten-orientiert und benötigen viele komplexe Transformationen für die Datenbewirtschaftung.

Für die Umsetzung des „noETL, yesSQL"-Verfahrens im Data Warehouse wird eine ELT-Engine benötigt. Das ist ein Tool oder ein Set von Tools, um Integrations-, Orchestrierungs- und Scheduling-Aufgaben in einem ELT-Prozess zu übernehmen. Dabei kann ein spezielles ELT-Tool zum Einsatz kommen oder ein Standard-ETL-Tool, das ELT-Verfahren unterstützt, oder es wird eine Kombination aus einem Datenintegrations-Tool und einem eigenen SQL-Framework verwendet *(siehe Abbildung 4)*.

Man unterscheidet grundsätzlich zwei Schritte in der Datenaufbereitung: Daten-Integration (EL) und Daten-Transformation (T). Im ersten Schritt werden die Daten aus den Quellen extrahiert und ins Data Warehouse geladen. Die Übertragung der Daten erfolgt „1:1" mit Datentyp-Konvertierung ins Zielformat. Für die Abfrage der Quellsysteme bieten sich SQL oder eine andere native Sprache des Quellsystems an (*Abbildung 4*, Punkt 1). Die Gestaltung des Daten-Integrationsschritts hängt von der jeweiligen Landschaft ab; die folgenden Optionen könnten einzeln oder in Kombination verwendet werden:

- Laden mit einem Integrations- oder Standard-ETL-Tool (*Abbildung 4*, Punkte 2a und 3a)
- Laden mit den nativen Integrations-Features der Datenbank (*Abbildung 4*, Punkte 2b und 3b)

Im zweiten Schritt werden die Daten im Data Warehouse mit SQL-Abfragen transformiert (*Abbildung 4*, Punkte 4 und 5). Bei der Entwicklung der Transformationen empfiehlt es sich, Plain-SQL für fachliche Transformationslogik zu verwenden. Einerseits ist die SQL-Ausführung in der Datenbank performant, weil sie für Massendaten-Operationen entworfen und optimiert wurde, andererseits vereinfacht sich Data Lineage, weil Plain-SQL besser geparst und analysiert werden kann. Um Performance und Wartbarkeit zu gewährleisten, sind die Transformationen möglichst granular zu halten. PL/SQL oder Scripting könnten zur Automatisierung der wiederkehrenden oder technischen Operationen eingesetzt werden. Um SQL-Wildwuchs zu vermeiden, bieten sich drei Optionen an:

- Beim Templating wird eine Vorlage erstellt, die die Entwickler verwenden können. Diese Templates können beispielsweise technische Routinen wie Logging beinhalten, damit sie nicht von jedem Entwickler neu ausprogrammiert werden müssen. Der Template-basierte Code der Entwickler wird anschließend ausgerollt. Die Nachteile des Templating bestehen darin, dass die falsche oder fehlerhafte Verwendung eines Templates schwer zu erkennen und zu verhindern ist. Zudem führen Änderungen an den Templates

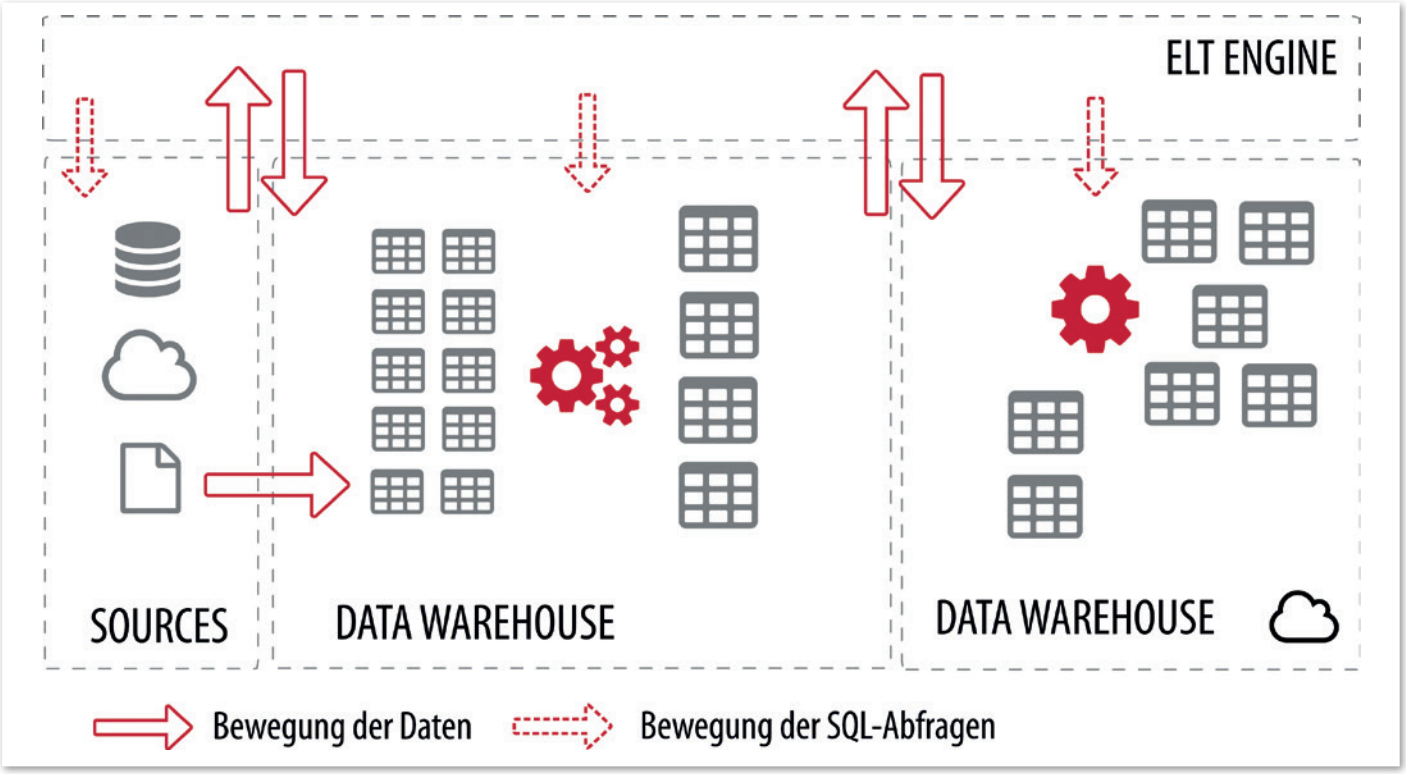


Abbildung 5: „noETL, yesSQL"-hybride Architektur im Data Warehouse

dazu, dass alle bereits laufenden Instanzen gegebenenfalls angepasst werden müssen.

- Code-Generierung erzeugt Instanzen anhand der Template-Vorlage und der Ergänzungen des Entwicklers. Die Generierung verhindert, dass der Entwickler das Template falsch oder unvollständig verwendet. Bei Änderungen des Templates ist eventuell das Re-Deployment aller bereits erzeugten Instanzen notwendig.
- Bei der Modularisierung wird der effektive Code zur Laufzeit anhand der im Modul hinterlegten Logik (Template) und des vom Entwickler eingegebenen Codes erzeugt. Bei Änderung der Modullogik ist lediglich das Re-Deployment des Moduls erforderlich. Das macht die Stärke dieser Methode bei der SQL-Entwicklung aus.

Idealerweise unterstützt die ELT-Engine eines dieser drei Verfahren. Der Aufbau einer zukunftssicheren Transformations-Architektur sollte von folgenden Überlegungen geleitet werden: Wo immer möglich, werden die Stärken der beteiligten Systeme ausgenutzt und die Transformationen an diejenige Umgebung delegiert (sogenanntes „Push-Down"), in der die Ausführung am effizientesten durchgeführt werden kann. Die „noETL, yesSQL"-Transformations-Architektur trägt genau diesem Kalkül Rechnung; dadurch ist sie transparent, flexibel und unabhängig von der transformierten Datenmenge, indem sie sich auf die Orchestrierung der Transformationen beschränkt. Auch in komplexeren Landschaften, in denen mehrere Systeme am Transformationsprozess beteiligt sind, bleibt der Grundgedanke der Transformations-Architektur derselbe, nur dass weitere Transformations-Plattformen hinzukommen. Dazu ein Beispiel, bei dem die Landschaft um eine weitere Cloud-Datenbank erweitert wird *(siehe Abbildung 5)*.

Bei diesem Aufbau kommt eine neue Integrationsstelle zwischen der On-Premises- und der Cloud-DWH-Datenbank dazu. Manche Cloud-Datenbanken bieten auch native Integrations-Komponenten an, die aus SQL heraus gesteuert werden können. Alles andere bleibt aus der Sicht der Transformations-Architektur identisch, die SQL-Transformationsabfragen werden lediglich an eine weitere Datenbank gesendet.

Fazit

In der modernen Data-Warehousing-Welt mit steigender Komplexität und wachsendem Daten-Aufkommen sind transparentere und flexiblere Datentransformations-Lösungen erforderlich. ELT ist ein alternatives Datentransformations-Verfahren, das sich besonders gut im Bereich von Massendaten-Verarbeitung für Data Warehousing bewährt hat.

Durch Anwendung des ELT-Verfahrens werden im Gegensatz zu ETL die Stärken der jeweiligen Datenspeicher-Technologien wie Cloud-, In-Memory-, MPP-Datenbanken, Hadoop oder Spark ausgespielt. Durch zahlreiche Features wie auf Massendaten optimierte Ausführungspläne, hohe Skalierbarkeit und reiche Integrationsfeatures bieten moderne Datenbanken durchaus eine passende Plattform für ELT.

SQL stellt dabei eine universelle, standardisierte und zukunftssichere Sprache für die Datenverarbeitung und Transformationsentwicklung dar. Die Kombination aus ELT und der Transformationsentwicklung mit SQL bietet eine sogenannte „noETL, yesSQL"-Transformationslandschaft. Diese benötigt nicht immer ein anderes Tooling, sondern vor allem eine andere Herangehensweise. Aus diesem Grund ist der Einstieg in die „noETL, yesSQL"-Welt auch schrittweise und ohne Big Bang möglich und sinnvoll.

Alec Shalashou
alec@datasqill.de

Noch mehr Flexibilität im Data Warehouse: Data Vault mit virtuellen Data Marts

Jörg Stahnke, PPI AG

Data Vault hat sich als flexibles Modellierungsverfahren für Data Warehouses (DWH) etabliert. Leider endet diese Flexibilität häufig bei der Erstellung der Data Marts. Zudem lassen sich Data-Vault-Modelle nur schwer abfragen. Anwender ohne tiefgreifende SQL-Kenntnisse verzweifeln nicht selten aufgrund der vielen Tabellen.

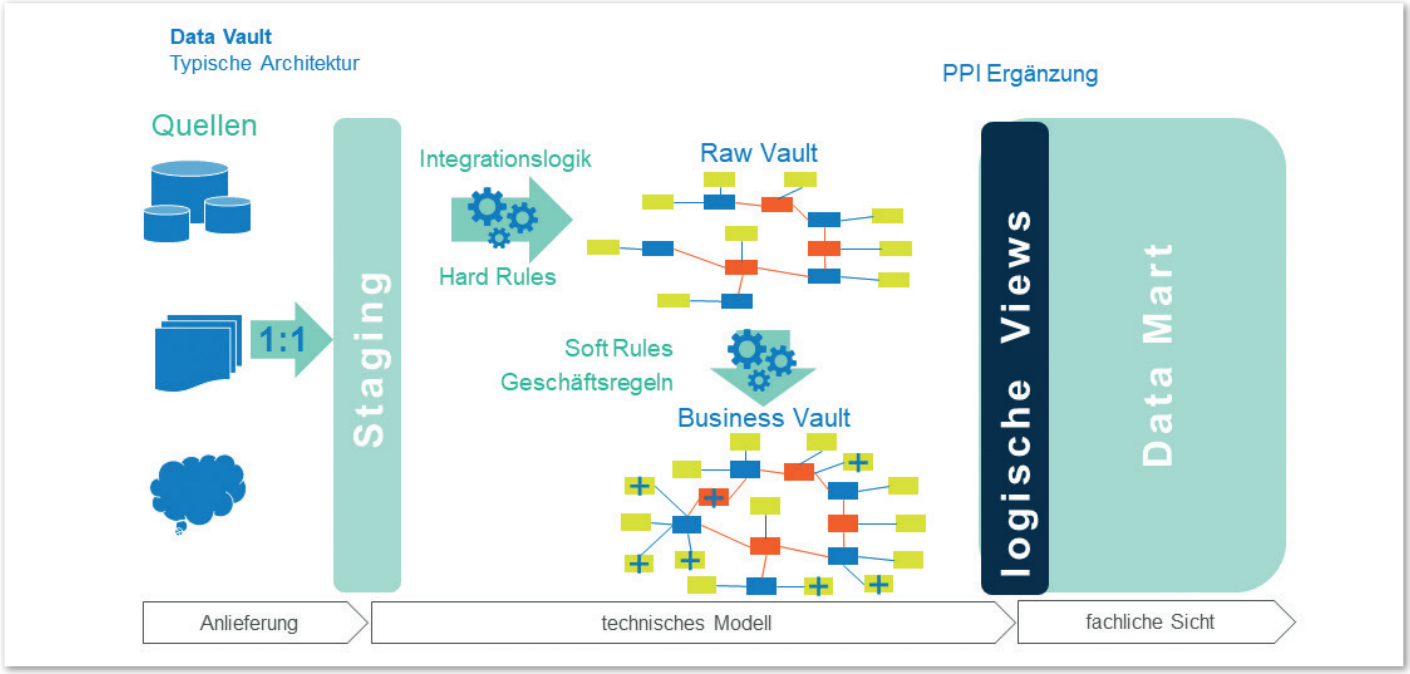


Abbildung 1: Data-Vault-Architektur mit logischen Views

PPI hat die Data-Vault-Standard-Architektur um eine virtuelle fachliche Zugriffsschicht (logische Views) ergänzt. Diese sind schnell zu implementieren, bieten leicht verständliche fachliche Sichten und gewährleisten durch konsequente Optimierung die notwendige Abfrage-Performance. Damit sind erste Voraussetzungen für ein vollständig virtualisiertes DWH der Zukunft geschaffen.

Abbildung 1 zeigt, dass beliebige Quellen ihre Daten unverändert in eine Staging-Schicht anliefern. Technische Hard Rules transformieren die Daten in ein Data-Vault-Modell (Raw Vault), ohne den Quellsystem-spezifischen Dateninhalt zu ändern. Typische Beispiele für Hard Rules sind Datumsformat-Konvertierungen oder Eindeutigkeitsprüfungen. Fachliche Soft Rules (Geschäftsregeln) wenden dann fachliche Logik an und konsolidieren die Quellsystem-spezifischen Daten in ein fachliches Data-Vault-Modell (Business Vault). Dabei werden bei Bedarf auch neue Kennziffern berechnet. Mithilfe der logischen Views wird dann eine fachliche Data-Mart-Sicht geschaffen, die Endanwender leicht auswerten können.

Die Architektur ist klar in die drei zielgruppenorientierten Bereiche „Anlieferung", „technisches Modell" und „fachliche Sicht" gegliedert:

- Die Staging-Schicht entspricht den Strukturen der Quellsysteme.
- Entwickler und Modellierer können im technischen Data-Vault-Modell alle Flexibilitätsvorteile des Data Vault nutzen und sehr effizient arbeiten.
- Fachanwender erhalten eine leicht verständliche fachliche Sicht; diese kann direkt für Auswertungen per SQL oder mit BI-Tools verwendet werden.

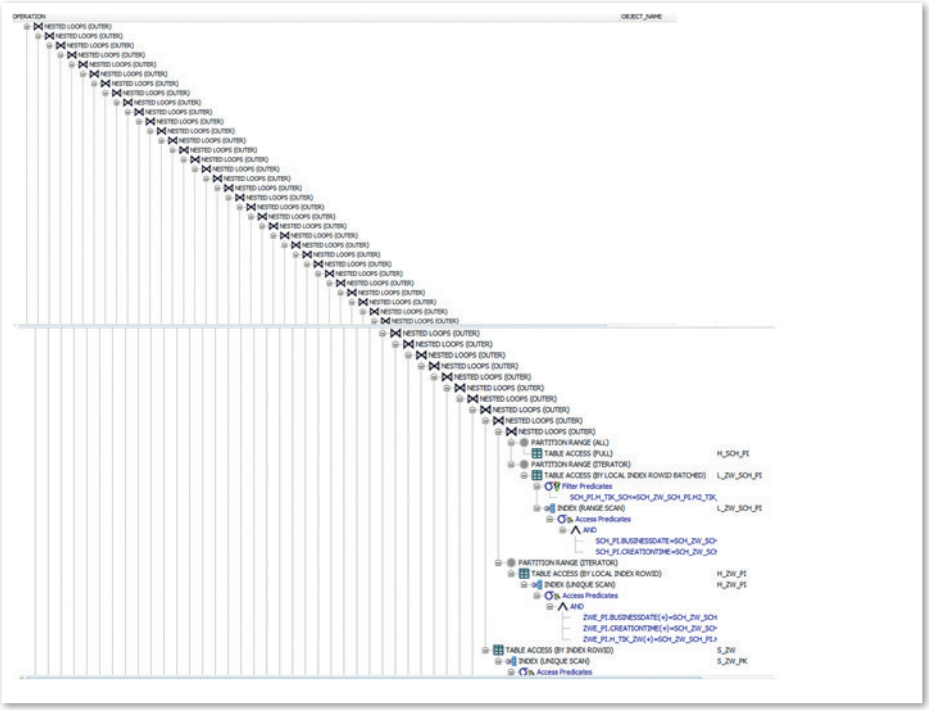


Abbildung 2: Ausführungsplan für Vollabfrage

Durch den Einsatz des von PPI entwickelten Universal Datamodel Generator (UDG) zur Data-Vault-Modellierung lassen sich auch die logischen Views innerhalb weniger Minuten modellieren. Die logischen Views bieten folgende Vorteile:

- Durch den geringen Erstellungsaufwand können verschiedene zielgruppenorientierte Views angelegt werden.
- Alle Views zeigen gleiche Grunddaten aus der Business Vault an, die Berechnungslogik ist also immer einheitlich. Dadurch sind inkonsistente Daten in verschiedenen Views/Data Marts ausgeschlossen.

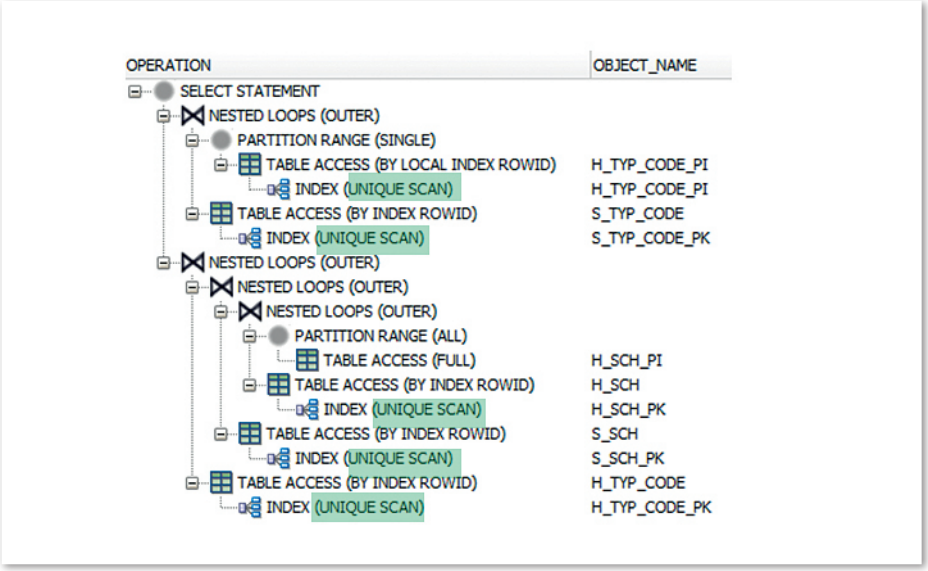


Abbildung 3: Ausführungsplan für übliche Auswertung

- Der Aufwand für Tests der Data-Mart-Dateninhalte ist extrem reduziert.
- Bei Data-Mart-Änderungen sind keine Datenmigrationen erforderlich.
- Der Fachbereich verfügt sofort über ein leicht verständliches fachliches Modell.
- Der Fachbereich benötigt keine spezifischen Data-Vault-Kenntnisse.

Performance logischer Views

Dem Einsatz logischer Views wird oft Skepsis hinsichtlich ihrer Performance entgegengebracht. Wenn die logischen Views jedoch konsequent die Optimierungsmöglichkeiten von Oracle nutzen, können sie performant abgefragt werden. PPI hat dabei positive Erfahrungen in Kundenprojekten gesammelt. Dabei waren folgende Punkte besonders wirkungsvoll:

- Ein ausreichend großer Datenbank-Cache
- Der Einsatz von Parallel Query
- Die Nutzung der Optimizer Join Elimination

Während die ersten beiden Punkte seit Jahren in vielen Oracle-Datenbanken zum Einsatz kommen, ist das mit Oracle 12.2 erweiterte Feature der Optimizer Join Elimination noch weitgehend unbekannt.

Optimizer Join Elimination

Die logischen Views stellen in der Regel sehr viele Felder bereit und nutzen bei der Abfrage eine große Anzahl von Tabellen (Hubs, Links, Satelliten und Point-in-Time-Tabellen). Eine typische Auswertung wertet aber nur wenige Felder aus. Die Optimizer Join Elimination ermittelt vollautomatisch, welche in der View verwendeten Tabellen tatsächlich zur Anzeige der in der konkreten Auswertung gewünschten Felder benötigt werden; auf alle anderen Tabellen wird nicht zugegriffen. Die überflüssigen Tabellen tauchen im Ausführungsplan gar nicht auf. Die folgenden Abbildungen verdeutlichen dies eindrucksvoll. Dabei wird die gleiche View („V_SCHADEN“) mehrfach abgefragt. Die Anzahl der ausgewählten Felder schwankt dabei.

Abbildung 2 zeigt trotz der kleinen Schriftgröße nur einen Ausschnitt des Ausführungsplans für „SELECT * FROM V_SCHADEN“. Es ist deutlich erkennbar, dass bei Abfrage aller Felder der View auf eine Vielzahl von Tabellen zugegriffen wird.

Abbildung 3 zeigt den vollständigen Ausführungsplan für „SELECT BUSINESSDATE, MANDANT_NUMMER, SOB, BETR_FG, SPERR_DT, SPER_FG, DECK_FG, AEND_DAT, OB, OB_TXT, PRIO_CD, PRIO_DE FROM V_SCHADEN“. Für die Abfrage dieser zwölf View-Felder muss nur auf sechs Tabellen zugegriffen werden. Auf eine Tabelle wird aufgrund der fehlenden Einschränkung in der „Where“-Klausel vollständig zugegriffen. Alle anderen Tabellen können unter Verwendung eindeutiger Indizes gelesen werden. Diese Abfrage ist der typische Anwendungsfall beim Einsatz von BI-Werkzeugen und wird durch die Optimizer Join Elimination performant ausgeführt.

Abbildung 4 zeigt den Ausführungsplan für „SELECT BUSINESSDATE FROM V_SCHADEN“. Bei Auswahl nur eines Feldes greift der Ausführungsplan nur auf einen einzigen Index zu. Natürlich ist diese Abfrage fachlich nicht sinnvoll. Sie zeigt jedoch, dass die Optimizer Join Elimination die überflüssigen Tabellen-Zugriffe wirklich vollständig wegoptimieren kann.

Das vollständig virtualisierte DWH der Zukunft

Virtualisierungen werden für einzelne DWH-Schichten bereits genutzt. PPI hat diese für die Data Marts erfolgreich eingesetzt. Roelant Vos stellte auf der Data Modeling Zone Konferenz 2017 vor, wie er die Raw Vault und Business Vault virtualisiert hat. Damit ist eine konsequente Virtualisierung über alle Schichten möglich. Gleichzeitig gibt es schon viele Projekte, die erfolgreich fachliche Logik mit Rule Engines abbilden. Durch den Übergang von programmierter Logik auf regelbasierte Logik lässt sich die Flexibilität weiter erhöhen. Daher stellt sich die Frage: „Kann man nicht alles miteinander kombinieren und ein vollständig virtuelles Data-Vault-Modell schaffen?“

Ein vollständig virtualisiertes DWH könnte sich die jeweils aktuellen Daten in einer temporären Landing Zone von den Quellen anliefern lassen und dann ohne jegliche Struktur-Änderung und Logik diese Daten in einer permanenten Staging-Schicht historisieren. Von dort werden die Daten virtuell in die Raw Vault und teilweise in die Business Vault übernommen. Fachliche Logik wird durch eine Rule Engine abgebildet und ist durch den Fachbereich jederzeit anpassbar. Auf der obersten Ebene repräsentieren die logischen Views die Data Marts *(siehe Abbildung 5)*.

Dieses Vorgehen bietet viele neue Möglichkeiten. Man kann nicht nur flexibel er-

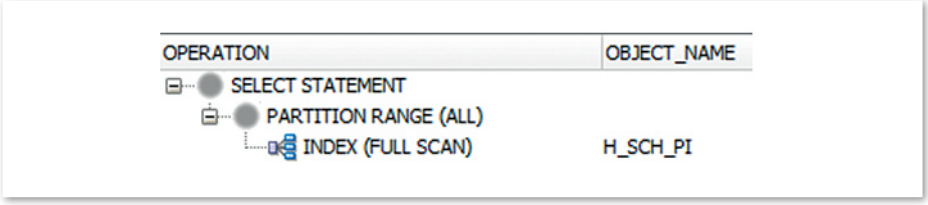


Abbildung 4: Ausführungsplan für Minimalabfrage

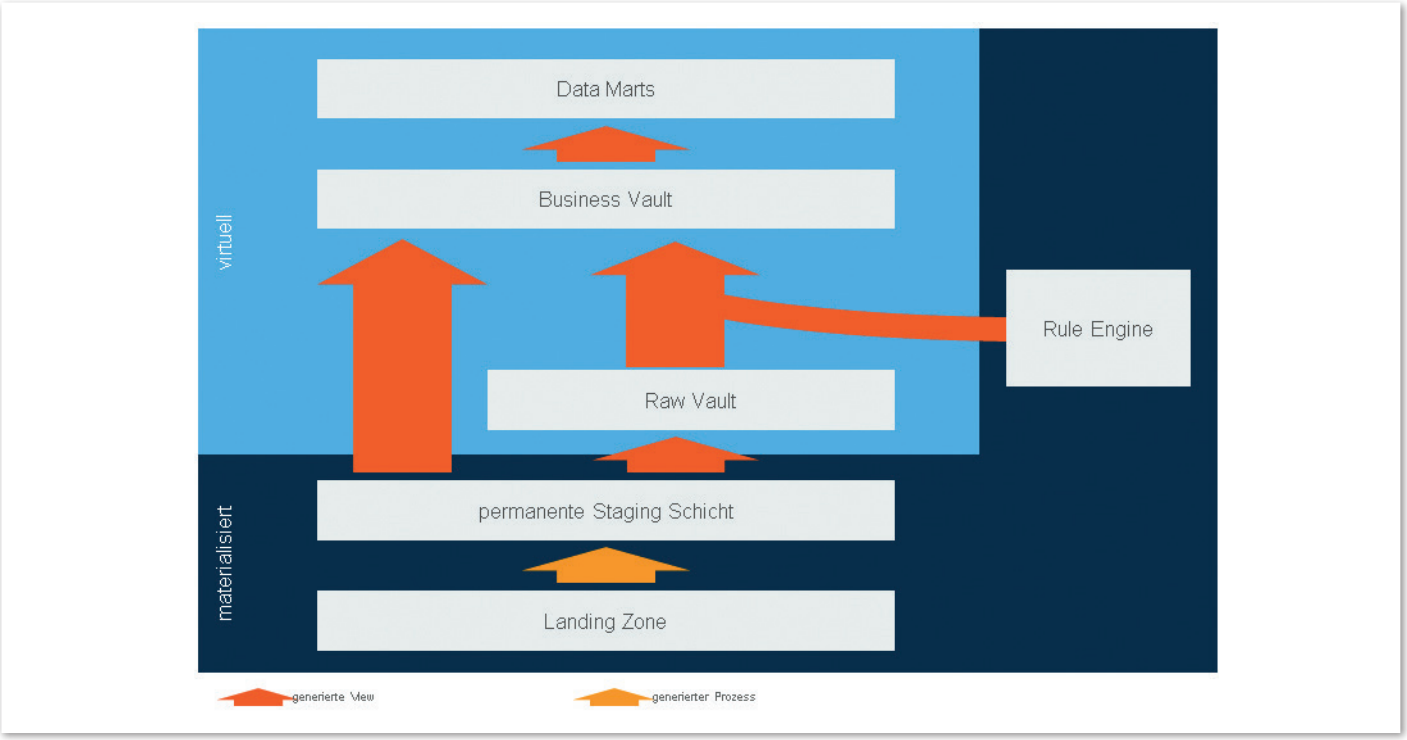


Abbildung 5: Architektur eines vollständig virtualisierten DWH

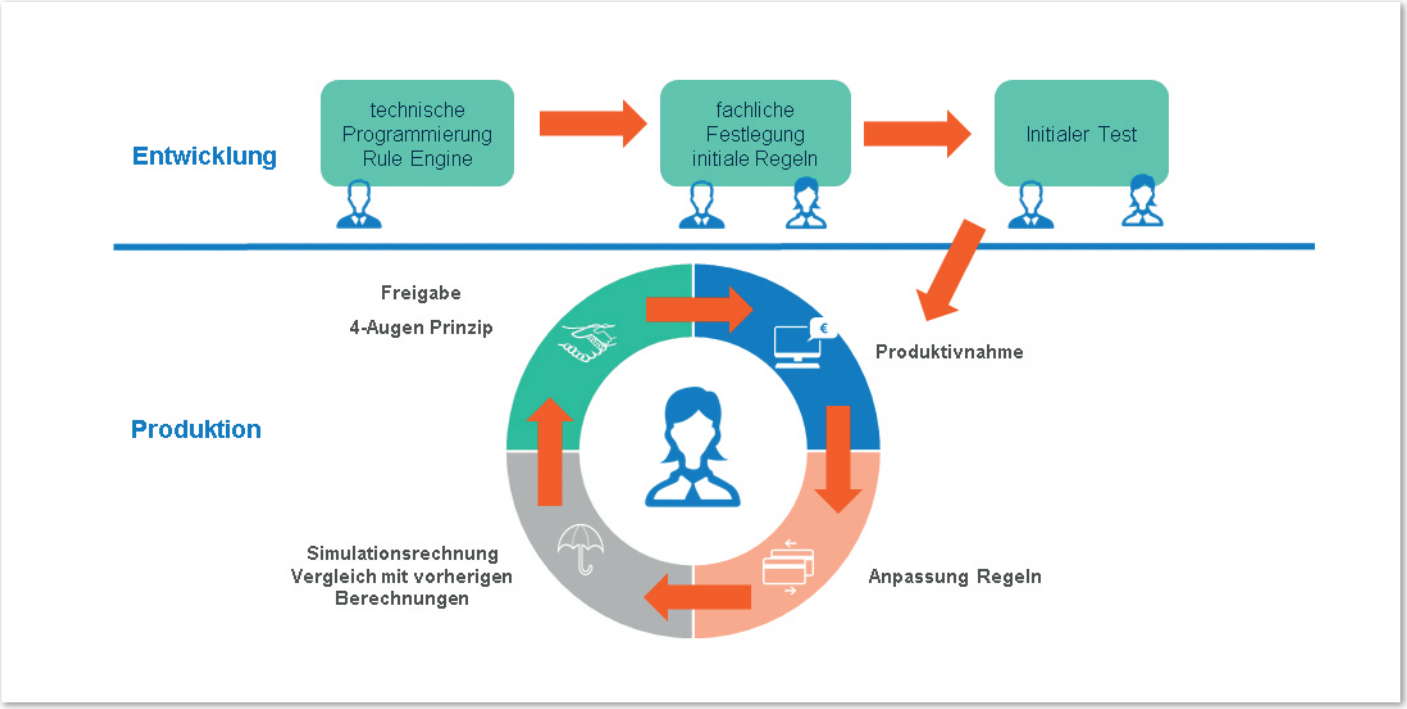


Abbildung 6: Vorgehensmodell mit Rule Engine

gänzen, sondern wirklich alles flexibel ändern. Die Fachabteilungen werden viel stärker einbezogen und können fachliche Änderungen sofort produktiv nehmen. Fachabteilung und IT vermeiden den klassischen Konflikt zwischen festen Releasezyklen und kurzfristig notwendigen fachlichen Anpassungen. Dieses Konzept bietet die folgenden offensichtlichen Vorteile:

- Sofortige Wirksamkeit neuer Daten und neuer Regeln ohne Ladelauf
- Reduzierter Speicherbedarf
- Weniger Daten-Redundanz
- Geringer Migrationsaufwand bei Änderungen

Daneben bieten sich aber auch völlig neue Möglichkeiten: So lassen sich etwa Quellsysteme sehr schnell bis zur permanenten Staging-Schicht anbinden. Historische Quelldaten werden schon gesammelt, während die Projektarbeit noch läuft. Bei Produktions-Einführung der Data-Vault- und Data-Mart-Modelle ist dann automatisch in Produktion bereits eine Historie vorhanden.

Man kann auch auf die gleiche Staging-Schicht zwei verschiedene virtuelle DWHs

setzen. Damit können beispielsweise zwei verschiedene Releases gleichzeitig aktiv sein. Dies ermöglicht beim Test direkte automatisierte Vergleiche der beiden Release-Stände. Beim Versionswechsel eines BI-Tools wäre ein gleitender Übergang von der alten auf die neue Version möglich. Neue Ideen können in einer Sandbox unabhängig vom regulären Produktionsbetrieb getestet werden. Verschiedene Niederlassungen könnten zu unterschiedlichen Terminen ihre individuellen Auswertungen an das neue Release anpassen.

Da keine umfangreichen Ladeläufe mehr erforderlich sind, kann mithilfe der Virtualisierung auch von täglichen Beladungen auf ein Near-Time- oder sogar auf ein Real-Time-DWH gewechselt werden. Der gleichzeitige Einsatz mehrerer virtueller Schichten mit den gleichen Grunddaten eröffnet somit vielfältige Einsatzmöglichkeiten.

Einsatz einer Rule Engine

Bei einer Rule Engine wird die fachliche Logik nicht in Quellcode programmiert, sondern als Regelsatz hinterlegt. Dadurch ist die IT dabei nur initial beteiligt. Die Regeln können durch den Fachbereich im laufenden Produktionsbetrieb angepasst und innerhalb eines Tages produktiv genommen werden. Dadurch lassen sich die in der Praxis häufig auftretenden kleineren Änderungen trotz fest geplanter Release-Zyklen für neue Software zeitnah durchführen.

Abbildung 6 zeigt die Vorgehensweise beim Einsatz einer Rule Engine. Zuerst muss die IT diese erstellen. Dabei sind folgende Fragen zu klären:

- Aus welchen Feldern sind die von der Rule Engine benötigten Informationen zu lesen?
- In welchen Feldern werden die Ergebnisse der Rule Engine abgelegt?

De facto beschäftigt sich der Entwickler hier mit dem Input-/Output-Verhalten der Rule Engine. Die letzten Details der fachlichen Regeln müssen zu diesem Zeitpunkt noch nicht bekannt sein. Anschließend erstellt die Fachabteilung gemeinsam mit der IT den initialen Satz an fachlichen Regeln. Diese werden im initialen Fachtest getestet. Bei Bedarf erfolgt hier eine Anpassung der Regeln.

Nach der Produktions-Einführung dieser Rule Engine inklusive des initialen Regelsatzes ist die IT an der Anpassung der Regeln nicht mehr beteiligt. Die Fachabteilung kann die Regeln im laufenden Produktivbetrieb anpassen (etwa neue Niederlassung beziehungsweise neue Produkte integrieren oder Sonderfälle einarbeiten).

Nach der Anpassung erfolgt eine Simulationsrechnung mit den geänderten Regeln. Die Ergebnisse werden mit den Ergebnissen der vorherigen Berechnung vollständig für alle Tabellen/Felder/Datensätze automatisiert verglichen. Der Fachbereich erhält einen vollständigen Überblick über alle Abweichungen und muss beurteilen, ob diese dem gewünschten Ziel entsprechen oder nicht. Je nach Entscheidung erfolgt dann eine Überarbeitung der angepassten Regeln mit erneutem Simulationslauf/Vergleich beziehungsweise die Freigabe der geänderten Regeln für die produktive Verwendung ab dem Folgetag im Vier-Augen-Prinzip. Die Revisionssicherheit und Nachvollziehbarkeit aller Änderungen ist durch dieses Prinzip gegeben.

Fazit

Virtualisierungen bieten viele neue, interessante Möglichkeiten; potenzielle Performance-Probleme kann man mit modernen Datenbank-Technologien in den Griff bekommen. Daher ist es sinnvoll, über diese Konzepte nachzudenken und sie in Projekten einzusetzen.

Jörg Stahnke
joerg.stahnke@ppi.de

Visualisierung von Geodaten in Oracle Apex

Alessandro Fondacaro, DB Systel GmbH

Reine Präsentation von Daten ohne visuelle Aufbereitung ist heute kaum vorstellbar. Grafische Darstellungen unterstützen ein schnelleres Verständnis – sowohl bei der Analyse wie auch bei der Datenpflege. Viele Daten lassen sich mit einem Bezug zur realen Welt versehen, indem sie um geometrische Informationen angereichert werden. Dieser Artikel stellt beispielhaft dar, wie diese Daten in eine Web-Anwendung integriert werden können und dem Bearbeiter eine komfortable Datenpflege ermöglichen. Die dabei eingesetzten Komponenten sind neben Oracle Application Express (Apex) die JavaScript-Bibliothek „OpenLayers" und die OpenSource-Software „Geoserver".

Zur Pflege von Daten einer Oracle-Datenbank bietet Oracle die Software-Entwicklungsumgebung Apex an. Sie ermöglicht es, in wenigen Schritten Oberflächen einer Webanwendung zu erzeugen und diese mit den Daten oder Funktionen der Datenbank zu verknüpfen. Die Entwicklung findet größtenteils im Browser statt. Für viele Aktionen sind dabei nur wenige Programmierkenntnisse erforderlich.

Die in Apex verfügbaren Komponenten werden auf der Oberfläche angeordnet und über Auswahllisten mit Funktionen versehen. Dabei kann SQL- oder PL/SQL-Code zum Einsatz kommen. Dies bietet unzählige Möglichkeiten, mit den Daten der Datenbank zu arbeiten. Mit HTML, CSS und JavaScript lässt sich die Anwendung an die eigenen Bedürfnisse anpassen – sowohl visuell als auch funktional. Für viele Datentypen reicht eine tabellarische Form zur Darstellung aus. Durch die Einbindung weiterer Komponenten können außerdem Daten mit einem geometrischen Bezug (Geodaten) in der Anwendung visualisiert und in den Prozess der Datenverarbeitung integriert werden.

Geodaten in Oracle

Damit Geodaten in einer Oracle-Datenbank persistiert und verarbeitet werden können, ist die Erweiterung „Oracle Locator" nötig. Sie beinhaltet die Kernfunktionen von „Oracle Spatial", die für den Anwendungsfall dieses Artikels ausreichend sind. Durch die Erweiterung steht für die Datenhaltung der zusätzliche Datentyp „SDO_GEOMETRY" zur Verfügung, der es erlaubt, die geometrischen Informationen des Datensatzes zu speichern.

Für den Umgang mit Geodaten gibt es Standards aus dem Open GIS Consortium (OGC). Die „Simple-Feature-Spezifikation" legt fest, welche Geometrie-Typen existieren müssen und wie diese in Bezug zueinander stehen. So kann die Position eines Bahnhofs als Punktgeometrie („Point") mit X- und Y-Koordinate hinterlegt werden. Die Strecke zwischen zwei Bahnhöfen wird durch einen Linienzug („LineString") definiert. Dieser Geometrie-Typ enthält einzelne Stützpunkte, die in der Darstellung mit Linien verbunden sind. Die Spezifikation legt auch Standards für die Repräsentation der Daten fest, etwa durch das Format „WKT" (Well Known Text). In diesem ist definiert, wie eine Geometrie in textueller Form beschrieben wird.

```
<head>
    [ol.css – Stylesheet]
    [ol.js – JavaScript-Bibliothek]
</head>

<body>
    [div-Element - Bereich für die Karte]
    [Script-Bereich - Inhalt und Funktionalitäten der Karte]
</body>
```

Listing 1

Da sich Oracle Spatial konform zur Simple-Feature-Spezifikation des Open GIS Consortium (OGC) verhält, ist die Konvertierung in oder aus anderen Formaten möglich. Zudem stehen die ebenfalls standardisierten Funktionen bereit, um Geodaten abzufragen oder zu verändern. Mit der Spatial-Erweiterung kann Oracle als vollwertige Geo-Datenbank genutzt werden.

Kartendarstellung mit OpenLayers

Die Standards für Web-Anwendungen entwickeln sich stetig weiter. Für die Darstellung und Funktionalität sind hier die Weiterentwicklungen von HTML zu HTML5, CSS zu CSS3 und JavaScript hervorzuheben. Dank der steigenden Browser-Unterstützung lassen sich mittlerweile viele dieser Neuerungen nutzen, ohne eine Anwendung auf bestimmte Web-Browser limitieren zu müssen.

Großen Zulaufs erfreuen sich außerdem Open-Source-Projekte. Dort findet eine gemeinsame Entwicklung an Projekten durch eine große Nutzergemeinde statt. Die Ergebnisse können offen eingesehen und genutzt werden. OpenLayers ist eines dieser Projekte. Es handelt sich dabei um eine JavaScript-Bibliothek, mit der eine interaktive Kartendarstellung im Web-Browser realisiert werden kann. Mit wenig eigenem Code kann eine Karte in ein HTML-Dokument eingefügt werden. Die umfangreiche Schnittstelle (API) sowie deren Dokumentation ermöglicht es, Karte und Funktionen an die eigenen Bedürfnisse anzupassen. *Listing 1* zeigt, an welcher Stelle die notwendigen Anpassungen in einem HTML-Dokument vorgenommen werden können, um eine Karte im Browser darzustellen.

Einbinden von OpenLayers in Apex

Um eine Karte mit OpenLayers in Apex darzustellen, müssen die einzelnen Fragmente an verschiedenen Stellen der Anwendung eingebaut werden. Zunächst ist auf der Seite ein Bereich zu definieren, in dem die Kartendarstellung erfolgen soll. Da in diesem Bereich HTML-Code eingefügt werden soll, wird hierzu eine Region vom Typ „STATIC_REGION" verwendet. Dieser Bereich wird dann vollständig mit der Karte gefüllt und von OpenLayers verwaltet.

Zusätzlich sind die notwendigen Bibliotheken in die Anwendung einzubinden. Apex stellt dafür bereits entsprechende Felder zur Konfiguration bereit. In den Page-Properties einer Seite gibt es die Möglichkeit, JavaScript- und CSS-Dateien einzubinden. Damit die Karte mit Inhalt gefüllt werden kann, muss im letzten Schritt der JavaScript-Code zur Konfiguration von OpenLayers in die Seite eingefügt werden. Für einen ersten schnellen Erfolg kann dieser ebenfalls in Apex hinterlegt werden (Page-Properties –> JS-Functions).

Für die bessere Wartbarkeit und eine komfortablere Entwicklung sollte der eigene JavaScript-Code ebenfalls ausgelagert und analog zur OpenLayers-Bibliothek eingebunden werden. Zu beachten ist, dass die Karte erst initialisiert werden sollte, nachdem die Apex-Anwendung vollständig geladen ist. Das vermeidet Probleme bei der Darstellung.

Darstellung der Geodaten über Standards

Zur Einbindung von geometrischen Daten in OpenLayers gibt es verschiedene Optionen. Eine Möglichkeit ist die Nutzung der OGC-Standards „Web-Map-Service" (WMS) und „Web-Feature-Service" (WFS). Bei WMS handelt es sich um eine Schnittstelle zur Abfrage von Geodaten in einem Rasterformat. Über HTTP wird ein Request gesendet, der die Information über den gewünschten Kartenausschnitt und die darzustellenden Daten beinhaltet. Das Ergebnis ist eine Datei in einem Rasterformat, etwa PNG oder JPEG. Diese kann dann in der Anwendung dargestellt werden. OpenLayers bietet die Funktion, einen WMS-Dienst einzubinden. Hierzu ist es

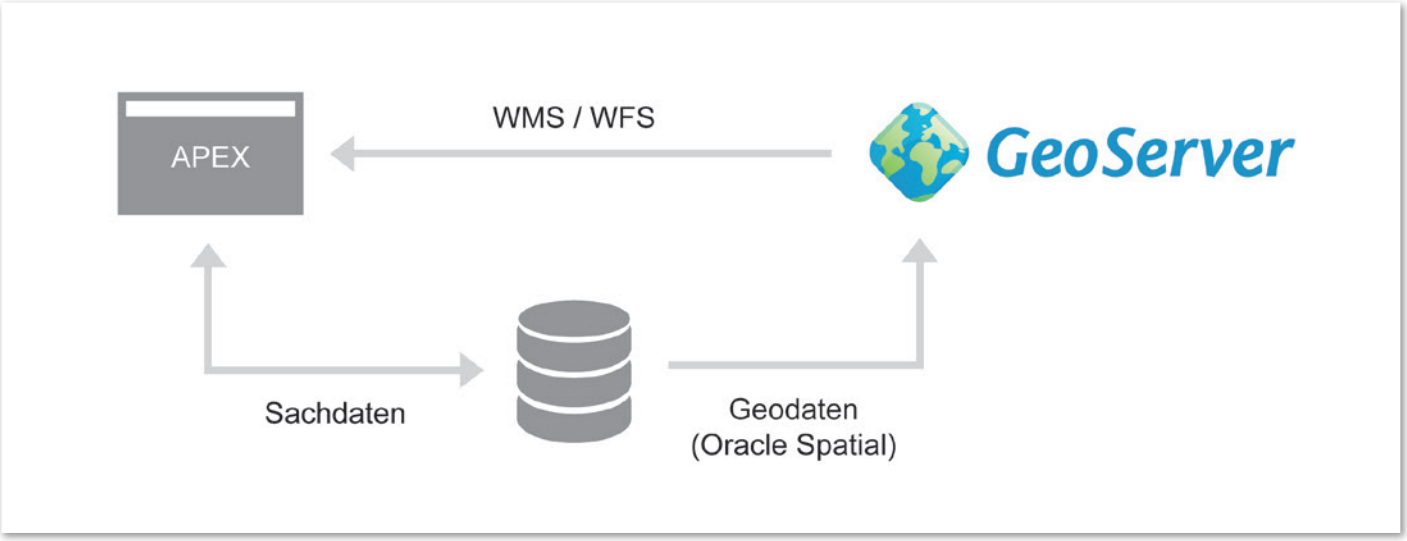


Abbildung 1: Die drei Komponenten einer Anwendung

```
 [Das Textfeld hat initial den Wert "Hallo"]
apex.item(<Apex-Item>).getValue() // Rückgabe "Hallo"
apex.item(<Apex-Item>).setValue("Welt")
[Das Textfeld hat nun den Wert "Welt"]
```

Listing 2

lediglich notwendig, die URL des Dienstes bekanntzugeben.

Für die Abfrage von Vektordaten wird der Web-Feature-Service verwendet. Hier wird ebenfalls ein HTTP-Request zu den geforderten Daten gesendet, das Ergebnis enthält jedoch einzelne Datensätze. Auch hier gibt es verschiedene Formate wie XML, GML, GeoJSON etc. Da OpenLayers es ermöglicht, Vektoren im Format GeoJSON einzubinden, ist dieses Format für die Abfrage eines WFS-Dienstes zu bevorzugen. Bei GeoJSON handelt es sich um eine Daten-Repräsentation im JSON-Format mit dem Zusatz, dass auch geometrische Informationen vorliegen.

Architektur der Anwendung

Die Open-Source-Software „Geoserver" kann Geodaten als WMS- oder WFS-Dienst bereitstellen. Es handelt sich dabei um eine auf Java basierende Anwendung, die mit einem Plug-in auf Oracle-Datenbanken zugreifen kann und in Oracle-Spatial vorliegende Daten als Dienst bereitstellt. Die Daten werden als abfragbare Layer angeboten. Diese lassen sich automatisch durch den Geoserver generieren oder auch durch eigene SQL-Statements erstellen. Für die Bereitstellung als WMS lassen sich sogenannte „Stile" in den Formaten CSS oder SLD konfigurieren. Die Vektordaten werden dadurch in Abhängigkeit von verschiedenen Bedingungen gemäß der Stilbeschreibung gezeichnet. Beispiele hierfür sind der Maßstab des angefragten Kartenausschnitts oder die Ausprägung eines Attributs.

Durch den Einsatz von Apex, OpenLayers und Geoserver lässt sich die Anwendung in drei Komponenten gliedern *(siehe Abbildung 1)*. Die Datenhaltung erfolgt in der Oracle-Datenbank. Dort sind alle Datensätze mit einer Geometrie und den geometrielosen Informationen versehen. Die Benutzeroberfläche ist in Apex implementiert. Dort ist auch OpenLayers für die Kartendarstellung eingebunden.

Die Daten der Apex-Komponenten werden direkt aus der Datenbank bezogen. Dies sind zum Beispiel die Inhalte der Auswahllisten oder Tabellen. Die darzustellenden Daten in der Karte werden durch OpenLayers über den Geoserver bezogen. Dieser greift wiederum auf die Oracle-Datenbank zu und stellt sie als Dienst bereit. Neben den eige-

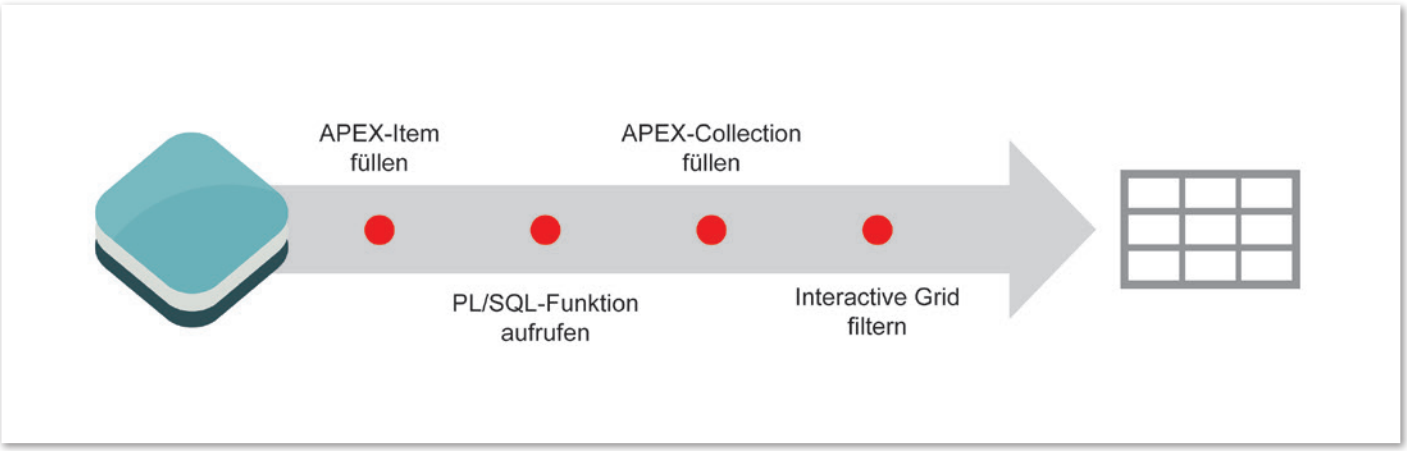


Abbildung 2: Das selektierte Element erkennen

nen Daten können auch fremde Dienste genutzt werden. Ein Beispiel ist die Darstellung einer Hintergrundkarte wie OpenStreetMap.

Interactive Grids zum Editieren der Daten

Die vielen Möglichkeiten, Daten über Standard-Komponenten in der Oberfläche zu präsentieren, ist ein großer Vorteil von Apex. Die Komponenten besitzen bereits Funktionen, die es dem Anwender erlauben, mit den Daten zu interagieren. Ein Interactive Report (IR) stellt die Daten tabellarisch dar und ermöglicht es dem Anwender, die Datenmenge zu filtern, neu zu sortieren oder bestimmte Datensätze hervorzuheben.

Ein neues Element in Apex 5 ist das Interactive Grid (IG). Dieses hat viele Komfortfunktionen des IR übernommen. Zusätzlich gestattet es das Editieren der Daten. Analog zu Programmen für Tabellenkalkulationen können die Werte in den Zellen verändert werden, außerdem kann man Auswahllisten hinterlegen. Der Anwender kann einen Wert aus der definierten Liste auswählen. Dadurch werden fehlerhafte Eingaben reduziert und die Datenqualität unterstützt.

Kommunikation zwischen Interactive Grids und OpenLayers

Alle wesentlichen Komponenten für die Anwendung sind nun beschrieben. Mit OpenLayers wird dem Anwender eine Karte zur Verfügung gestellt, in der alle erforderlichen Daten visualisiert sind. Der Anwender soll die Möglichkeit haben, ein Objekt in der Karte zu selektieren und die Daten des selektierten Objekts in einer Tabelle darzustellen, um sie dort zu pflegen. Diese Darstellung erfolgt in einem Interactive Grid. Die Herausforderung besteht nun darin, das selektierte Element zu erkennen und nur diese Daten im IG anzuzeigen. *Abbildung 2* stellt die notwendigen Schritte dar.

Apex stellt eine JavaScript-Bibliothek bereit, die verschiedene Funktionen für die Interaktion mit Apex-Komponenten beinhaltet. Ein Objekt, das für die Kommunikation genutzt wird, ist „apex.item". Es besitzt Funktionen, um den Wert einer Apex-Komponente auszulesen oder zu setzen. *Listing 2* zeigt ein einfaches Beispiel, um mit einem Textfeld zu interagieren.

Diese Funktionen ermöglichen es, den Inhalt der Karte durch andere Apex-Komponenten zu steuern. Zum Beispiel können Objekte ein- und ausgeblendet oder ein Zeitraum für den darzustellenden Datenstand definiert werden. Der Wert wird im JavaScript-Code ausgelesen und als Parameter in die WMS- oder WFS-Abfrage übergeben.

```
SELECT * FROM sachdatentabelle
WHERE id IN
(SELECT id FROM V_COLL_<Name der Collection>)
```

Listing 3

```
Apex_COLLECTION.ADD_MEMBER(
p_collection_name => <Name der Collection>,
p_c001 => <ID>)
```

Listing 4

Für das Füllen des Interactive Grid wird ein weiteres neues Feature von Apex verwendet: Apex-Collections. Einfach beschrieben handelt es sich hierbei um temporäre Tabellen. Jeder Benutzer hat dadurch seinen eigenen Zwischenspeicher, da dieser pro Session vorgehalten wird. In den Tabellen werden die Informationen zur Selektion hinterlegt, damit das Interactive Grid nur die Datensätze darstellt, die durch den Anwender in der Karte selektiert wurden. Das SQL-Statement, das dem IG zugrunde liegt, filtert die Datenmenge bereits auf die Datensätze, die in der Apex-Collection hinterlegt sind *(siehe Listing 3)*. Die Apex-Collection wird mit PL/SQL-Funktionen angelegt und gefüllt. *Listing 4* zeigt ein Beispiel, um einen Datensatz hinzuzufügen.

Die Ermittlung der ID ist abhängig von den dargestellten Daten. Sofern die Daten direkt als Vektor eingebunden sind, ist die ID bereits in OpenLayers bekannt, da sie als Attribut durch den WFS geliefert wurde. Werden die Daten über WMS eingebunden, was lediglich einer Bilddatei entspricht, kann beim „GeoServer" angefragt werden, welches Element sich an der Position der Selektion befindet.

Um den Wert in die PL/SQL-Prozedur zu übernehmen und die Apex-Collection zu füllen, kann er zunächst über „apex.item" in eine Apex-Komponente geschrieben werden. Ein verstecktes Textfeld ist dafür geeignet; es wird mit einer „DynamicAction" versehen. Mit dieser können Bedingungen definiert werden, die beim Zutreffen eine Aktion ausführen.

Für das Übernehmen der ID in die Apex-Collection lautet die Bedingung, dass sich der Wert geändert hat. Trifft dies zu, wird die PL/SQL-Prozedur ausgeführt und die neue ID in der Apex-Collection hinterlegt. Anschließend muss die Aktualisierung des Interactive Grid angestoßen werden. Dieses filtert den kompletten Datenstand auf die hinterlegten IDs, die wiederum der aktuellen Selektion entsprechen.

Fazit

Durch die neuen Funktionen und Elemente in Apex 5 ist es mit geringem Aufwand möglich, eine optisch ansprechende und moderne Web-Anwendung zu erstellen. Die Interactive Grids gestatten eine komfortable Bearbeitung der Daten. Da durch JavaScript auch weitere Komponenten wie OpenLayers eingebunden werden können, lassen sich Apex-Anwendungen erweitern, um auch Geodaten visualisieren zu können und den Anwender mit diesen Daten interagieren zu lassen. Grundlage dafür sind die mit Apex ausgelieferten JavaScript-Bibliotheken. Daher ist es wünschenswert, dass diese auch weiterhin gepflegt und erweitert werden.

Alessandro Fondacaro
alessandro.fondacaro@deutschebahn.com

Datenqualitäts-Cockpit zur Analyse und Steuerung der Datenqualität

Christiane Breuer, Christian Haag und Alexander Jochum, DATA MART Consulting GmbH

Durch immer neue und stetig wachsende regulatorische Anforderungen an Kreditinstitute und Finanzdienstleistungsunternehmen gerät das Thema „Datenqualität" – auch im Hinblick auf die Zunahme der Datenmengen – in den Fokus. Es führt dazu, dass immer mehr BI-Manager in die Verantwortung genommen werden, die Datenqualität auch zu attestieren. Der Artikel zeigt, wie man sinnvolle Kennzahlen und Messgrößen auf Basis eines nach Data Vault 2.0 konzipierten Data Warehouse schafft, mit derer Hilfe man die Daten- und Prozessqualität visualisieren und darstellen kann, und illustriert die Möglichkeiten der Oracle Business Intelligence Enterprise Edition (OBIEE). Darüber hinaus sind die in diesem Zusammenhang entstandenen Herausforderungen und Schwierigkeiten erläutert.

Die regulatorischen Anforderungen an Kreditinstitute und Akteure an den Finanzmärkten sind in den letzten Jahren enorm gestiegen; ein Trend, der sich in den nächsten Jahren fortsetzen wird. Für diesen Use Case ist dargelegt, warum ein Datenqualitätsmanagement notwendig ist, wozu ein Datenqualitäts-Control-Framework dient und wo die Managementaufgaben liegen. Anschließend geht es um die Reifegrade der Qualitätsmanagement-Prozesse und darum, wie die Qualität in diesem Kontext definiert ist.

Die Qualitätsattribute in Informations-Systemen lassen sich in der DQ-Matrix kompakt darstellen. Zur Abbildung der Datenqualität in den ETL-Prozessen bedarf es eines Daten- und Prozessmodells. Dies illustrieren das Beispiel „Architektur- und Prozessmodell" sowie das vereinfachte DQCF-Datenmodell. Im zweiten Teil sind das konkrete Beispiel einer Implementierung auf einem Data Warehouse (Oracle 12c/Data Vault 2.0) und das Beispiel einer Visualisierung mit OBIEE beschrieben.

Der Sinn des Datenqualitätsmanagements

Die Aufgabenstellung und vor allen die Erwartungshaltung an BI und Data Warehouse sind zunächst einmal durch die Endanwendersicht bestimmt. Betriebswirtschaftliche Inhalte sollen durch BI-/BA-Tools und entsprechende Funktionalität bereitgestellt werden. Demgegenüber stehen die operativen Vorsysteme und Geschäftsprozesse, deren strukturierte und unstrukturierte Daten es anzubinden gilt. Beim Data-Lake- und/oder Data-Warehouse-Ansatz geht es nun darum, die Daten und Strukturen zu modellieren, die Prozesse abzubilden, die Daten zu synchronisieren und ihre Qualität sicherzustellen – kurz: um die Informationsbereitstellung *(siehe Abbildung 1)*.

Die Aufwandsverteilung in einem typischen Data-Warehouse-/BI-Projekt stellt sich hingegen umgekehrt dar. Während typischerweise nur zehn bis fünfundzwanzig Prozent des Aufwands auf Auswertung, Berichtswesen, Analyse (BI-/BA-Frontends) entfallen, liegt der Großteil des Implementierungsaufwands mit fünfundsiebzig bis neunzig Prozent im Bereich der Datenbewirtschaftung, also bei der Datenmodellierung, Extraktion, Transformation, Compliance sowie beim Laden, Verarbeiten und Datenqualitätsmanagement *(siehe Abbildung 2)*.

Deshalb ist das Bild des Eisberges so einprägsam – die Datenbewirtschaftung und das Datenqualitätsmanagement sind zwar der Hauptteil eines Implementierungsprojekts, vor allem bei der Integration mehrerer Datenquellen, aber im Idealfall für den Benutzer nicht sichtbar.

Das Datenqualitäts-Framework

Datenqualität muss messbar sein; dokumentierte Datenqualität schafft Sicherheit und Vertrauen in Informationen. Aber auch legale Anforderungen zum Sign-off der Finanz-Informationen bedingen bereits die Einführung eines Datenqualitäts-Frameworks, Beispiele sind der Sarbanes Oxley Act (SOX 302/SOX 404) oder Basel Committee on Banking Supervision (BCBS 239).

Die Aufgaben des Datenqualitätsmanagements

Hier ist zwischen technischen, fachlichen und organisatorischen Prozessen zu unterscheiden, und das sowohl bereits in der Projektierung als auch im Betrieb. Bei den technischen Prozessen bestehen die Aufgaben des Datenqualitätsmanagements darin, Datenqualität als integrativen Teil der Datenbewirtschaftungsprozesse abzubilden. Dabei müssen sie zwingend automatisiert, protokolliert, messbar und idealerweise selbstheilend umgesetzt sein. Beispiele dafür, wie die Datenqualität als Teil der Datenbewirtschaftungsprozesse umgesetzt werden kann, sind:

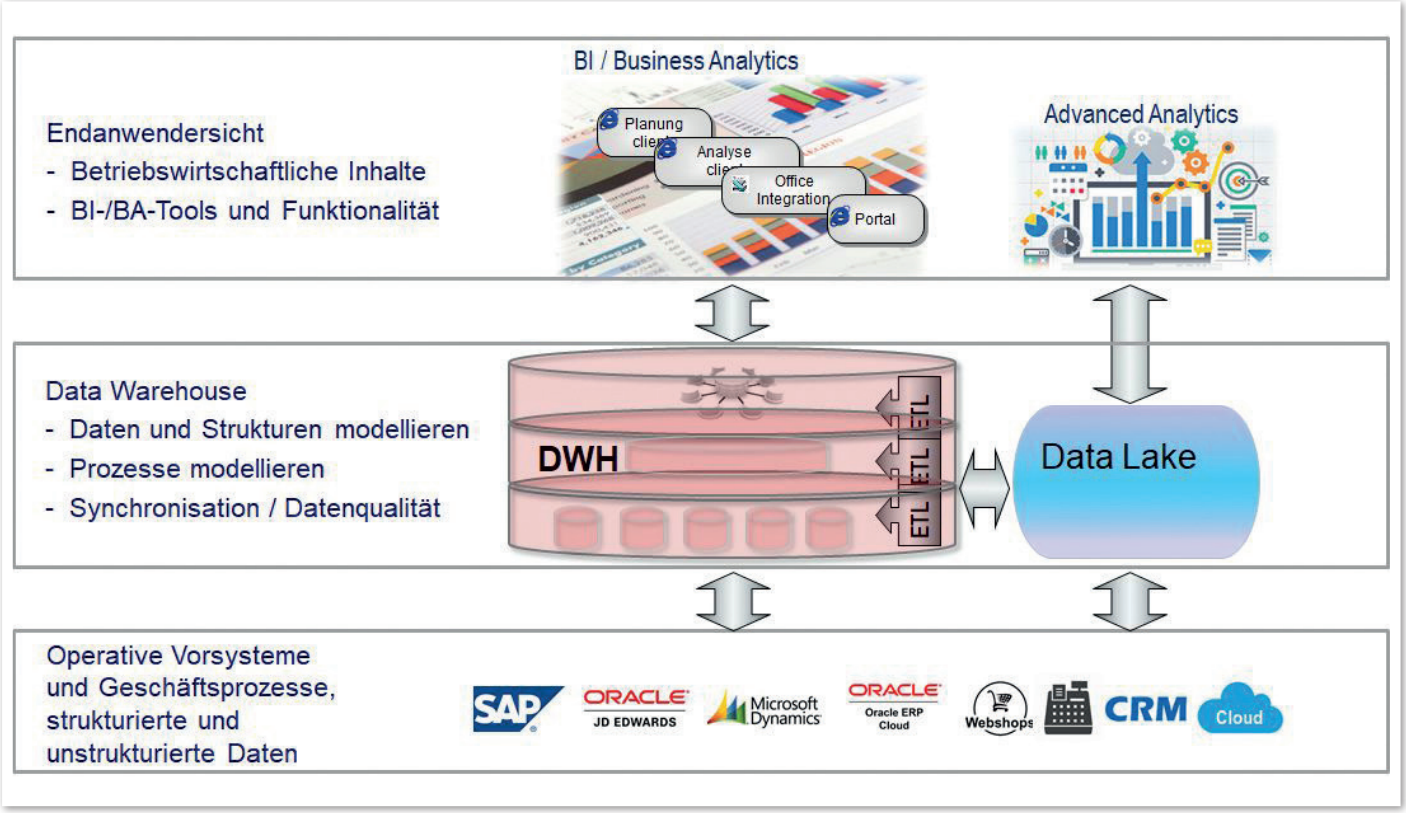


Abbildung 1: Aufgabenstellung/Erwartungshaltung an BI und Data Warehouse

Abbildung 2: Aufwandsverteilung in einem typischen Data-Warehouse-/BI-Projekt

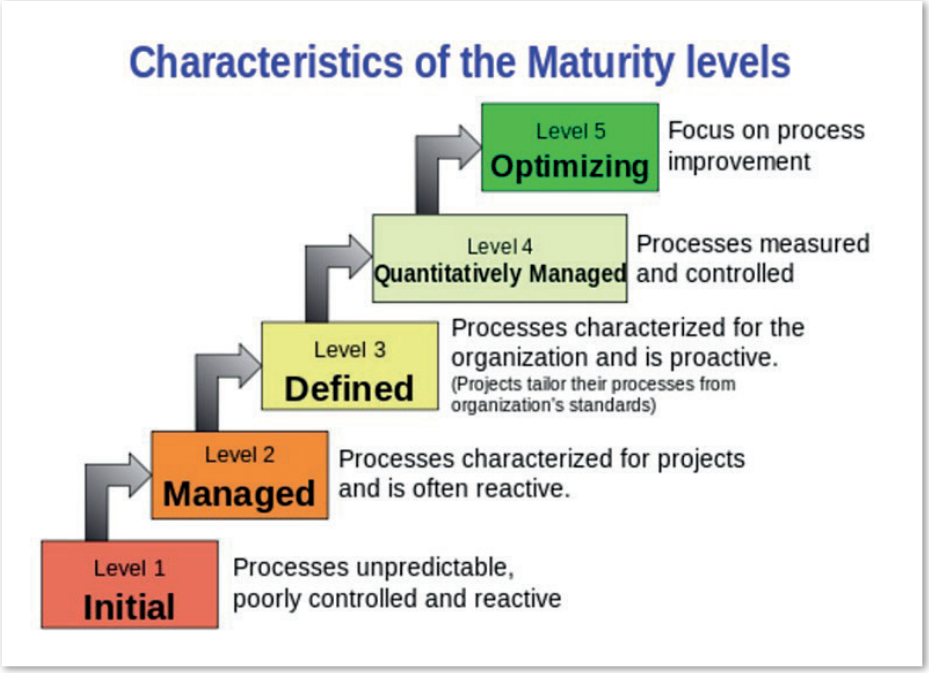


Abbildung 3: Reifegrade der Qualitätsmanagement-Prozesse

- Syntaktische (technische) Prüfungen
  - Technische Satzprüfungen (Feldtypen, Längen, Inhalte, Schlüssel etc.)
  - Vollständigkeitsprüfungen (Schnittstellenebene, Delta-Load-Kriterien, Statistik etc.)
  - Aktualitätsprüfungen (date range loaded, Delta-Load, change-dates etc.)

- Semantische (inhaltliche) Prüfungen
  - Strukturprüfungen (Synchronisation, Transformation, Stammdatenentsprechung)
  - Zuordnungsprüfung/Konsistenz (Historisierung, veränderliche Strukturen)
  - Validität (inhaltliche Prüfungen, Business Rules)
  - Korrektheit (Prüfsummenverfahren, Parallelprüfung, Datenabgleich etc.)

- Prozessmonitoring/Alerting (IT-Betrieb)

Die Aufgaben innerhalb der fachlichen Prozesse beinhalten initiale und laufende Datenqualitäts-Prüfungen, aufgeteilt nach der fachlichen Verantwortlichkeit. Dabei ist die Spezifikationsqualität entscheidend, also die Beschreibung der erforderlichen inhaltlichen DQ-Prüfungen, der Testverfahren mit Referenzangaben und der Business-Logik. Beispiele für konkrete Aufgaben sind:

- Datenbewirtschaftung/DWH-Modellierung
  - Erstellung von DQ-Kennzahlen und DQ-Datenmodell
  - DQ-Informationen als „Push"-Verfahren
  - Laufende semantische Prüfungen (Validierung, Plausibilität, Stammdatenprüfung etc.)

- Fachliche Aufgaben
  - Spezifikationen definieren in Projekt und Betrieb
  - Eindeutige Kennzahlen-Definitionen mit Business-Rules
  - Festlegung der DQ-Kriterien und DQ-Prüfungen
  - Initiale und laufende DQ-Prüfung gemäß Verantwortung (Plausibilität)
  - Unterstützung durch Visualisierung im BI-Tool (Prüfberichte, Exception-Berichte, Datenqualitätscockpit)

Bei den organisatorischen Prozessen gilt es, die Verantwortung für die beschriebene fachliche und technische Seite zu definieren, weiterhin für Betrieb, BICC Business Intelligence Competence Center und weitere organisatorisch relevante Bereiche. Dann sind die Prozesse der Spezifikation, der Umsetzung, der Testverfahren und der laufenden Kontrolle zu erarbeiten. Dabei werden Aufgaben wie Change-Management, Release-Management, Dokumentation und DQ-Pflichtkriterien beschrieben. Beispiele für organisatorische Aufgaben des Datenqualitätsmanagements sind:

- Dedizierte Organisation (BICC, DWH/BI-Team) aufbauen
  - mit Mitarbeitern aus IT und Fachbereichen
  - Schnittstellenfunktion und Anforderungsaufnahme
  - Sichern Spezifikation, Designqualität, Test, Abnahme

- DWH/BI als Unternehmensprozess etablieren und Prozesse definieren für
  - Anforderungs-/Erweiterungsmanagement
  - Fachlichen und technischen Support
  - Change-Management und Release-Management

Reifegrade der Qualitätsmanagement-Prozesse

Das Ergebnis dieser Aufgaben ist schlicht „Qualität", doch wodurch sind die Reifegrade der Qualitätsmanagement-Prozesse bestimmt? Eine Definition liefert das Capability-Maturity-Model-Integration-Modell (CMMI) (Basis: 1979 von Philip B. Crosby, Quality Management Maturity Grid (CMM), CMMI 1.0: US Ministry of Defense (2000–2002), CMMI 1.3: SEA 2010, Software Engineering Institute, *siehe Abbildung 3)*.

Nach Erfahrung der Autoren ist bereits im Projekt anzustreben, vom Reifegrad 1 (ad-hoc, ungeplant, als Reaktion im Fehlerfalle) über den Reifegrad 2 (intuitiv, sporadische Prüfungen, fokussiert auf Einzelpersonen, kaum Prozesskontrolle, unvollständig) und den Reifegrad 3 (qualitativ, statische QM-Maßnahmen, Prozesse und Verantwortli-

che sind definiert und institutionalisiert) bis zum Reifegrad 4 (quantitativ, messbare und kontrollierte Prozesse etabliert, Prozesserfahrung, automatisiert) zu gelangen. Konsequent gelebt, kann man dann im Betrieb auf dem Reifegrad 4, zumindest aber bei 3 aufsetzen und den höchsten Reifegrad 5 ansteuern (kontinuierliche Prozessverbesserung, strukturgetrieben (dynamisch), Monitoring zur Schwachstellen-Erkennung).

Die Qualität definieren

Bei der Definition ist die Qualität der Informationen, Prozesse, Vorgaben und Umsetzung zu unterscheiden. Qualität der Daten und Informationen ist gegeben, wenn die Qualitätsattribute für Daten getroffen werden. Die Qualität der Prozesse lässt sich an Zuverlässigkeit, Protokollierung, Dokumentation und Audit der Prozesse messen. Zudem sind Verfügbarkeit, Wartbarkeit und Nachvollziehbarkeit hier wichtige Kenngrößen.

Die Qualität der Definitionen und Vorgaben ist an der Spezifikationsqualität und an der Definition der DQ-Kriterien festgemacht. Die Qualität der Umsetzung schließlich ist bestimmt durch Messbarkeit, Kontrolle/Maßnahmen, unterstützende Organisation und Reviews. Als Instrumente dienen Zieldefinitionen, Messung der Zielerreichung und Verantwortung. Die Qualität eines BI-Systems kann daran gemessen werden, wie gut die Lösung zur schnellen, verständlichen und zuverlässigen Beantwortung der jeweiligen betriebswirtschaftlichen Fragestellungen geeignet ist.

Die DQ-Matrix

Mit Qualitätsdimensionen für DQ-Kennzahlen werden wir konkreter, um Qualitätsmängel fassbar und tatsächlich auch analysierbar zu machen und dadurch die Ableitung von Maßnahmen zu erleichtern. *Tabelle 1* zeigt gängige Qualitätsattribute beziehungsweise Qualitätsdimensionen dafür.

Ein Beispiel einer solchen DQ-Matrix ist es, die DQ-Dimensionen in den Spalten darzustellen (wie Anzahl Quellsystemfehler, fehlende Ladelose, unvollständige Ladelose, abweichende Ladelose, Staging-Abweichung, Hist-Abweichung, Analyse-Abweichung, Stammdatenfehler, Plausibilitätsfehler, Fehler in der Business-Logik, Fehler-Anreicherungen, Prüfsummen-Fehler, Fehler in der Cube-Verarbeitung) und die DQ-Kennzahlen in den Zeilen:

- Stammdaten-Verarbeitung
  - DQ-Kunde/Debitor
  - DQ-Kunde/CRM
  - DQ-Organisation
  - DQ-Mandant
  - DQ-Produkt
  - DQ-Lieferant/Kreditor
  - DQ-Kostenstelle
  - DQ-Kostenart
  - DQ-Sachkonten
  - DQ-Verträge
- Fakten-Verarbeitung
  - DQ-Fakturadaten
  - DQ-CRM-Daten
  - DQ-Mailing-Daten
  - DQ-WEB-Tracking
  - DQ-Vertragsdaten
  - DQ-Konditionen
  - DQ-FiBu Salden
  - DQ-FiBu Anlagen

Die Werte im Bericht ergeben sich dann aus den Zählungen der Fehler bei der jeweils letzten Beladung oder summiert über verschiedene Wochen- oder Monats-Zeiträume.

Die Protokollierung der Prozesse und der Prüfungen erfolgt auf der Ebene der technischen Bewirtschaftungsprozesse (hier per ODI auf Oracle), angereichert um die Ergebnisse jeder einzelnen Qualitätsprüfung in den Einzelschritten. In den Zeilen werden die einzelnen ETL-Prozesse mit ihrem Namen, Startdatum, Server, JobID, Start- und Ende-Zeitstempel, Laufzeit, Lademodus DELTA/FULL, Event, Result, #records source, #records target und Message aufgeführt. Daneben stehen die Spalten mit der Anzahl der Systemfehler, System-Warnings, Stammdatenfehler, Plausibilitätsfehler etc. Über die Auswahl des Betrachtungstags, Einschränkung der DWH-Ebene, Auswahl eines Jobs oder Beschränkung nur auf die Fehlerfälle kann gefiltert und eine Detailanalyse ermöglicht werden.

Datenqualität in den ETL-Prozessen

Für eine zuverlässige Datenbewirtschaftung und das zugehörige Datenqualitätsmanagement sind sowohl ein Daten- als auch ein Prozessmodell zu definieren. Das Prozessmodell enthält definierte Prozesse in standardisierten Prozessebenen (ETL), während das Datenmodell definierte Fehler- und Datenqualitäts-Tabellen umfasst, die ebenso wie die Finanz- oder Controlling-Daten für die Verantwortlichen aufzubereiten sind.

Das Monitoring der Datenqualität hat ebenfalls zwei Perspektiven: Die systematische Überwachung der Prozessqualität beantwortet die Frage: „Wie sind die Prozesse gelaufen?“ Das Monitoring der Datenqualität hingegen gibt Auskunft darüber, ob die Daten inhaltlich stimmen.

Beispiel Architektur- und Prozessmodell

Die nachfolgende Darstellung eines beispielhaften Architekturmodells zeigt den DWH- und BI-Aufbau im Kontext des Datenqualitätsmanagements. Jeder Schicht (von den Quellen/Vorsystemen über die Sta-

Qualitätsattribute/-dimensionen	Qualitätsprüfungen und Bildung von Qualitätskennzahlen (Beispiele)
Vollständigkeit der Daten	Prüfung Quellsysteme, Abweichung zum Durchschnitt, Datensätze Quellvergleich etc.
Fachliche Korrektheit der Zahlen	Fehleranzahl Validierung, fehlende Regeln, fehlende Anreicherungen etc.
Technische Korrektheit der Zahlen (mathematisch)	Abweichungen Prüfsummen (pro Ebene), Datentyp-Fehler, Feldfehler, NULL-Prüfungen etc.
Konsistenz (Übereinstimmung)	Anzahl Stammdatenfehler, Anzahl Historisierungsfehler, Anzahl Zuordnungsdopplungen etc. (jeweils pro Fact/fachlicher Dimension)
Struktur-Integrität (Interpretierbarkeit, Granularität)	Anzahl Transformationsfehler, Anzahl unlogischer Veränderungen etc.
Aktualität	Aktualisierung Quellsystem, Aktualität Ladeprüfung, Umfang Delta-Load

Tabelle 1

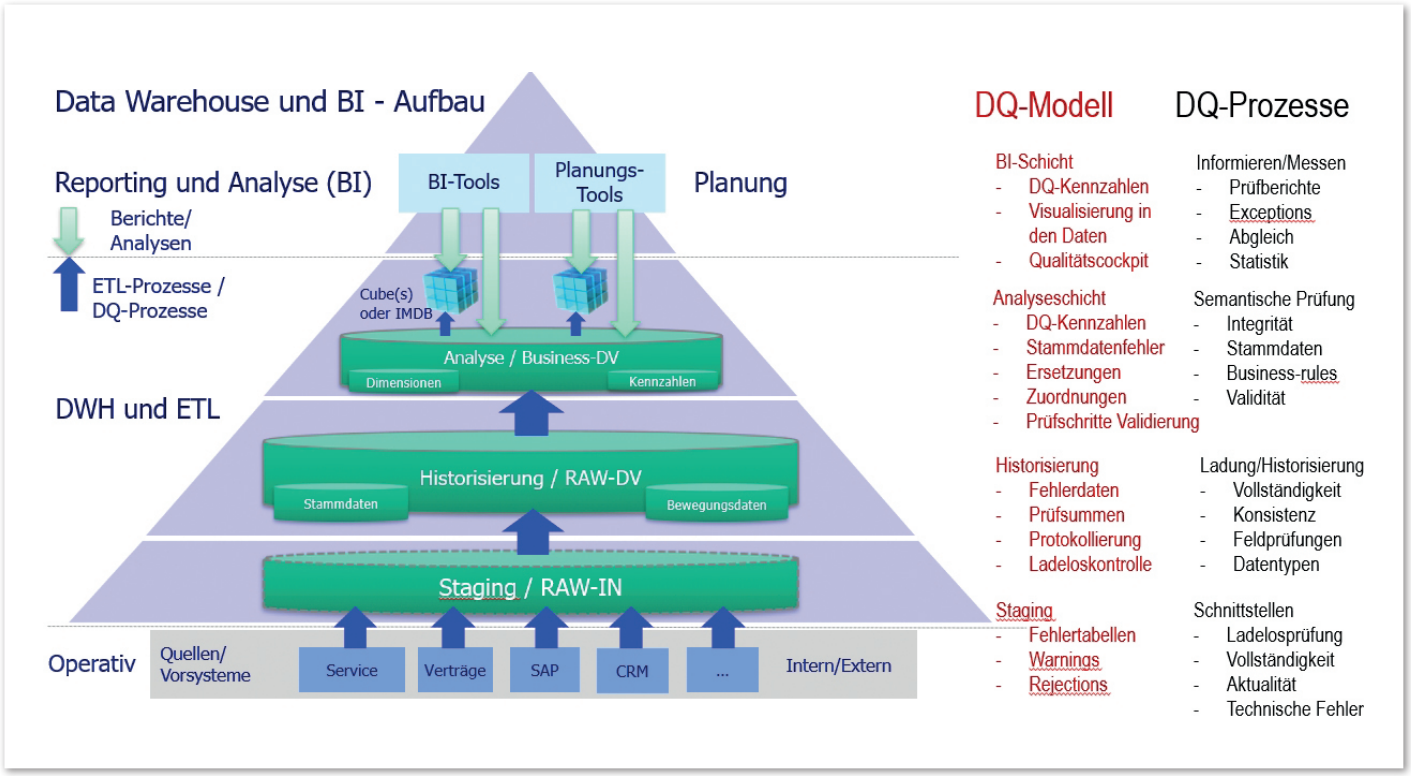


Abbildung 4: Beispiel Architektur- und Prozessmodell

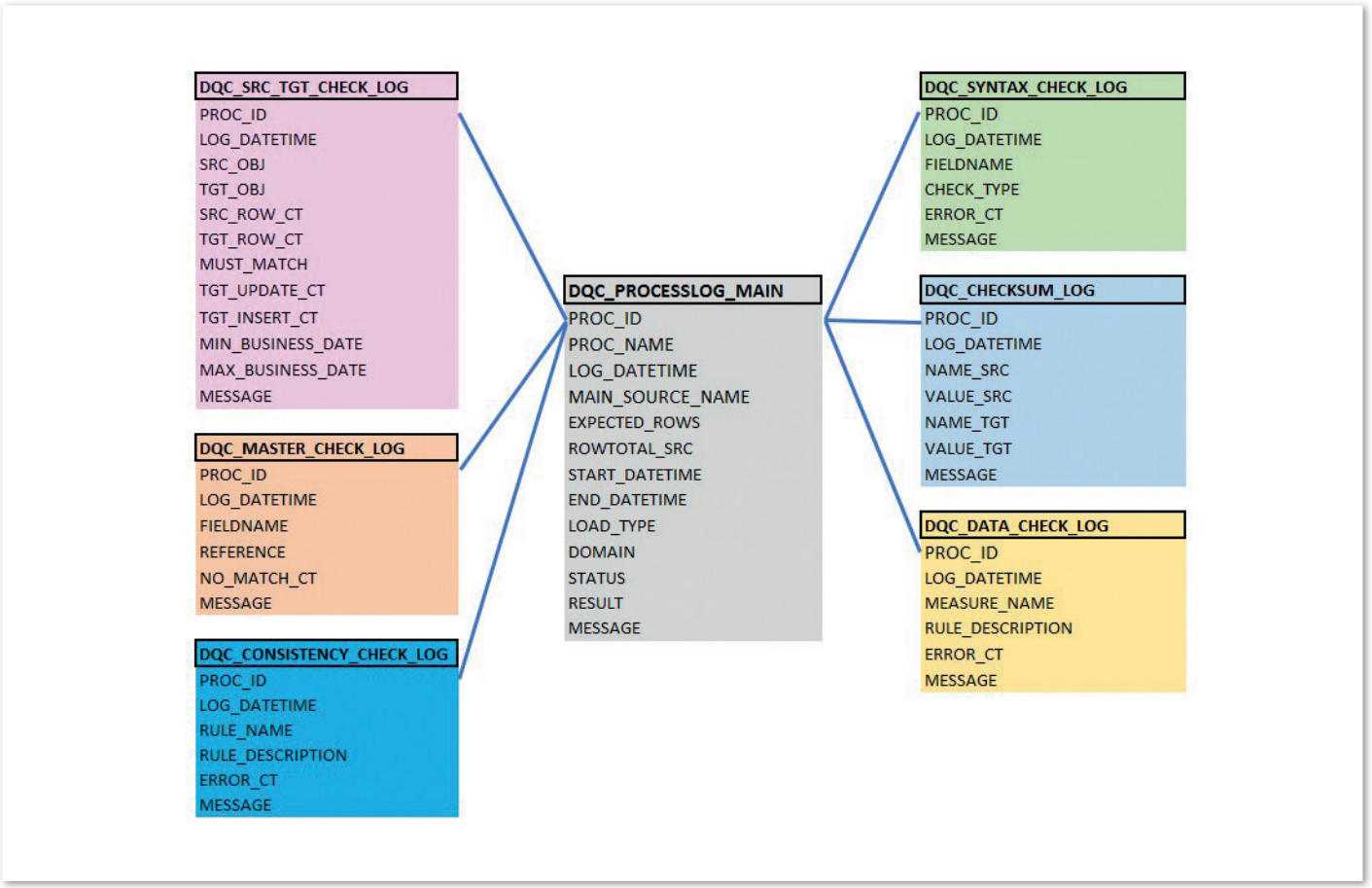


Abbildung 5: Vereinfachtes DQCF-Datenmodell

ging-/Raw-In-Schicht, die Historisierungs-/Raw-Data-Vault-Schicht und Analyse-/Business-Data-Vault-Schicht bis hin zur BI-Schicht) sind die Komponenten des DQ-Modells und der DQ-Prozesse zugeordnet *(siehe Abbildung 4)*.

Das Data Quality Control Framework (DQCF) im konkreten Projekt sollte nicht nur universell einsetzbar sein, sondern auch unabhängig von den eingesetzten ETL-Tools zur Datenbewirtschaftung des DWH (Oracle Data Integrator, PL/SQL, DataStage, WhereScape, SAS etc.) modelliert werden. Daher wurde ein vereinfachtes DQCF-Datenmodell mit dem Ziel entworfen, alle DQ-Prozesse in ein gemeinsames DQ-Modell zu protokollieren *(siehe Abbildung 5)*. Die zentrale Tabelle „DQC_PROCESSLOG_MAIN" enthält dabei die Basis-Informationen pro Prozess, während in den sechs weiteren damit verknüpften Tabellen die Details protokolliert werden *(siehe Abbildung 6)*.

Beispiel einer Implementierung auf einem DWH

Die nachfolgend beschriebene Umsetzung des DQCF basiert auf einer Oracle-Plattform. Als Datenbank kommt Oracle 12c (12.1) zum Einsatz, die Visualisierung ist mit OBIEE 12c (12.2) realisiert. Das Data-Warehouse-Design ist als Data Vault 2.0 mit den Schichten „Sources", „Raw-In", „Raw Data Vault", „Business Data Vault" und „Information Mart" modelliert. Besondere Herausforderungen sind dabei die große Anzahl Tabellen von Raw Data Vault nach Business Data Vault und das nicht persistente Business Data Vault. Bei der OBIEE-Implementierung sind folgende Besonderheiten zum Einsatz gekommen:

- Repository Highlights
  - Dynamic row-wise Initialization Variables used for Measure Tooltips
  - Writeback for Tooltip Maintenance
  - Dynamic Connection Pool (Different Databases: DEV, UAT, PRD)

- Visualization Highlights
  - Master-Detail-Reports
  - Conditional Drill-downs
  - Report Pop-ups
  - Conditional Formatting based on complex calculations

Im OBIEE Repository wurden die drei erforderlichen Layer (Physical Layer, Business Layer und Presentation Layer) modelliert. Das Business Model wurde als modifiziertes Star Schema entworfen. Sogenannte „Dynamic Variables" wurden im OBIEE-Repository mit einer Abfrage für dynamische Tool-Tipps definiert. Dabei erfolgt eine row-wise initialization. Der HTML-Code „<b title="@{biServer.variables['NQ_SESSION.Variable1']}">Measure 1<b>" wird dazu in den Analysis-Spalten-Überschriften hinterlegt.

Damit die Analysten in der Oberfläche zwischen den verschiedenen Umgebungen „Development", „UAT" und „Produktion" umschalten können, wurde ein Dynamic-Connection-Pool konfiguriert. Ein Prompt mit DSN-Selektion übergibt die Auswahl (DEV, UAT, PRD) an die Repository-Session-Variable. Der Eintrag für die dynamische Connection-Pool-Selektion lautet „VALUEOF(NQ_SESSION.DSN)".

Beispiel einer Visualisierung mit OBIEE

Als Beispiele für die Visualisierung folgen nun ausgewählte Beschreibungen der Berichte und Dashboards. Zu jeder DQ-Kennzahl wurde ein Master-Detail-Report erstellt, der intuitiv als Drill-down-Bericht aufzurufen ist. Im Master-Detail-Bericht erfolgt ein Context-Filtering auf die im selben Bildschirm sichtbare Grafik.

Der Analyst wird aktiv geführt, hier am Beispiel eines bedingten Drill-downs: Der Drill-down wird nur im Fehlerfalle aktiv und nur bei der Analyse bestimmter Ebenen. Beim Kontext-Drill-down über die rechte Maustaste erscheint ein Kontext-Mini-Report als Pop-up im Daten-Kontext und fokussiert somit auf die relevanten Details.

Verschiedene OBIEE Dashboards wurden implementiert, beispielsweise eine DQ-Matrix mit einem eindeutigen Color Coding (einschließlich Bewertung/Benotung) über die DQ-Dimensionen, aufgegliedert nach fachlichen Themen, und daneben die DQ-Kennzahlen mit den Fehlerquoten.

Das „Duration"-Dashboard stellt die Laufzeit für jeden einzelnen ETL-Prozess seiner durchschnittlichen Dauer der letzten dreißig Tage gegenüber. Dies kann nach Prozess-Level, fachlichen Themen, Gesellschaft, Umgebung sowie Zeiträumen gefiltert und verdichtet ausgewertet werden

Im Dashboard „Process Overview" basiert die bedingte Formatierung der Kennzahl „Prozess-Status in Prozent" auf komplexen Berechnungen mit unterschiedlichen Regeln je nach Prozess-Schicht. So kann eine Null-Fehler-Toleranz beispielsweise für das Laden der Rohdateien gelten, während

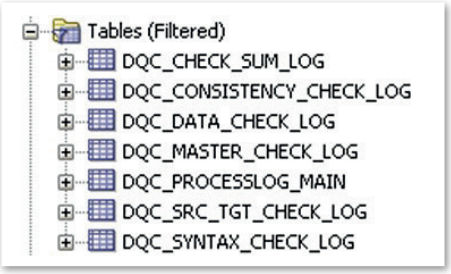


Abbildung 6: Die sieben Data-Quality-Control-Tabellen

für den Layer vom RDV zum BDV andere Schwellwerte herangezogen werden. Als letztes Beispiel sei der „Delivery"-Status genannt. Er zeigt, ob die erwarteten Quellen gemäß Liefer-Pattern bereitgestellt wurden, sowie deren fehlerlose Verarbeitung.

Fazit

Die Rolle der Datenqualität in der Entscheidungsunterstützung hat verschiedene Aspekte. Zum einen bildet sie das Fundament der Systeme zur Entscheidungsunterstützung. „Payback" (Amortisation) von Investition in Datenqualität wird durch Wartbarkeit, niedrigere Betriebskosten und sichere Entscheidungsgrundlagen erreicht. Die Formel für die Qualität in der Entscheidungsunterstützung lautet: gesicherte Datenqualität der Geschäftsdaten plus intelligente Strukturierung und Anreicherung der Daten plus Qualität der Datenpräsentation und -analyse. Datenqualität ist unsichtbar – wenn sie vorhanden ist …

Christiane Breuer
c.breuer@datamart.de

Geballte Ladung Applications für Oracle-Anwender und -Experten auf der DOAG 2018 Konferenz + Ausstellung

Dr. Frank Schönthaler, Leiter der Business Solutions Community

Networking, Praxiserfahrung, Expertenwissen, Diskussionen, Referate … die DOAG 2018 Konferenz + Ausstellung wird neue Entwicklungen eingehend thematisieren. Die Teilnehmer erhalten in bewährter Weise neben Fachdiskussionen echte Erfahrungsberichte und unaufgeregte Expertenmeinungen, um dem fachlichen Diskurs die notwendige Würze zu verleihen. Auch in diesem Jahr wird die Business Solutions Community wieder mit von der Partie sein.

Eine gehörige Portion Digitalisierung ist heute an jeder Ecke zu bekommen: Moderne Technologien, Big Data, Industrie 4.0 und nun auch Artificial Intelligence und Machine Learning beherrschen die Medien weltweit und die Hersteller werden nicht müde, die Segnungen dieser Technologien zu preisen und im Markt eine digitale Torschlusspanik zu erzeugen.

Dirk Blaurock, Applications-Themenverantwortlicher für den Applications Track auf der DOAG 2018 Konferenz + Ausstellung, hat mit seinem engagierten Organisationsteam ein neues Konzept für die Veranstaltung entwickelt. Neben bewährten Anwenderbeiträgen sind drei Workshops geplant, um Fokusthemen intensiv zu behandeln. Sie bieten kurze Impuls- und Erfahrungsberichte sowie anschließend viel Raum für die aktive Mitwirkung des Auditoriums an der fachlichen Diskussion. Themen lassen sich so passgenau und in optimaler Intensität behandeln. Folgende Workshop-Themen sind geplant:

- Oracle-Lizenzen bei Business-Applikationen: Wie ist der Stand, was ist zu beachten?
- Rest-API-Integration in die Oracle E-Business Suite
- Stammdaten-Management: Enabler und Treiber der digitalen Transformation

Keine Frage: Oracle-Applications-Anwender und -Experten können sich auf eine geballte Ladung Applications gefasst machen. Dafür stehen die Oracle-Keynote-Speaker Dr. Thomas Bruggner, Vice President Cloud Applications, und Dr. Nadia Bendjedou, Vice President Applications Development, die beide die Oracle-Applications-Strategie aus deutscher und internationaler Sicht transparent machen werden. Daneben wird sich Dr. Nadia Bendjedou, DOAG Botschafterin 2016, in ihrem Fachvortrag mit Top-Themen wie „GDPR", „Move to the Cloud", „Koexistenz in hybriden Umgebungen" und „Adaptive Intelligent Apps" befassen. Sie wird in diesem Jahr von Jacques Bouchet begleitet, der bei Oracle weltweit für die Lokalisierung der Unternehmensapplikationen verantwortlich zeichnet. Selbstverständlich sind beide offen für kritische Fragen und werden sicher wertvolle Hinweise und Tipps in ihrem Reisegepäck haben.

Die Business Solutions Community erwartet Sie vom 20. bis 23. November 2018 auf der DOAG 2018 Konferenz + Ausstellung in Nürnberg und freut sich auf einen interessanten Meinungs- und Erfahrungsaustausch. Weitere Informationen unter *„https://2018.doag.org"*.

Dr. Frank Schönthaler
frank.schoenthaler@doag.org

Data Analytics 2019

Die Datenexplosion meistern

26. & 27. März | Phantasialand bei Köln

analytics.doag.org

ORACLE Cloud

DOAG

20. - 23. November in Nürnberg

2018.doag.org

PROGRAMM ONLINE
mit rund 450 Vorträgen

Eventpartner: AOUG SOUG swiss oracle user group iJUG Verbund ORACLE