

II.3 Erfolgreiche Datenvalidierung und -filterung

Die Verwendbarkeit der Daten zu prüfen und fehlerhafte Daten auszufiltern – das sind die Aufgaben von Validierung und Filterung. Im Folgenden werden die verschiedenen Ebenen und Elemente der Validierung genauso dargestellt wie die Verfahren der Filterung. Weiterhin wird diskutiert, welche Vor- und Nachteile eine Validierung bei der Extraktion und beim Laden hat und wie man die verschiedenen Verfahren sinnvoll kombiniert.

Die Validierung weist nach objektiven Gesichtspunkten nach, ob sich die Daten für den vorgesehenen Zweck verwenden lassen. Die nicht verwendbaren Daten mit unzureichender Datenqualität werden dabei ausgefiltert. Man kann auf vier Ebenen validieren und filtern: der Attributs-, der Datensatz-, der Datenobjekt- und der Bestands-Ebene.

II.3.1 Validierung auf vier Ebenen

Auf **Attributs-Ebene** werden einzelne Datenelemente einer Struktur, beispielsweise eines Datensatzes, validiert und gefiltert. So wird z.B. geprüft, ob das Attribut `EINSTELLUNGS_DATUM` in dem Datensatz für einen neuen Mitarbeiter mit einem validen Datum gefüllt ist. In der Praxis erfolgt die Validierung eines Attributs üblicherweise mit mehr als einer Regel. Beispiele für diese Regeln sind in Abschnitt II.2.3.1 aufgeführt.

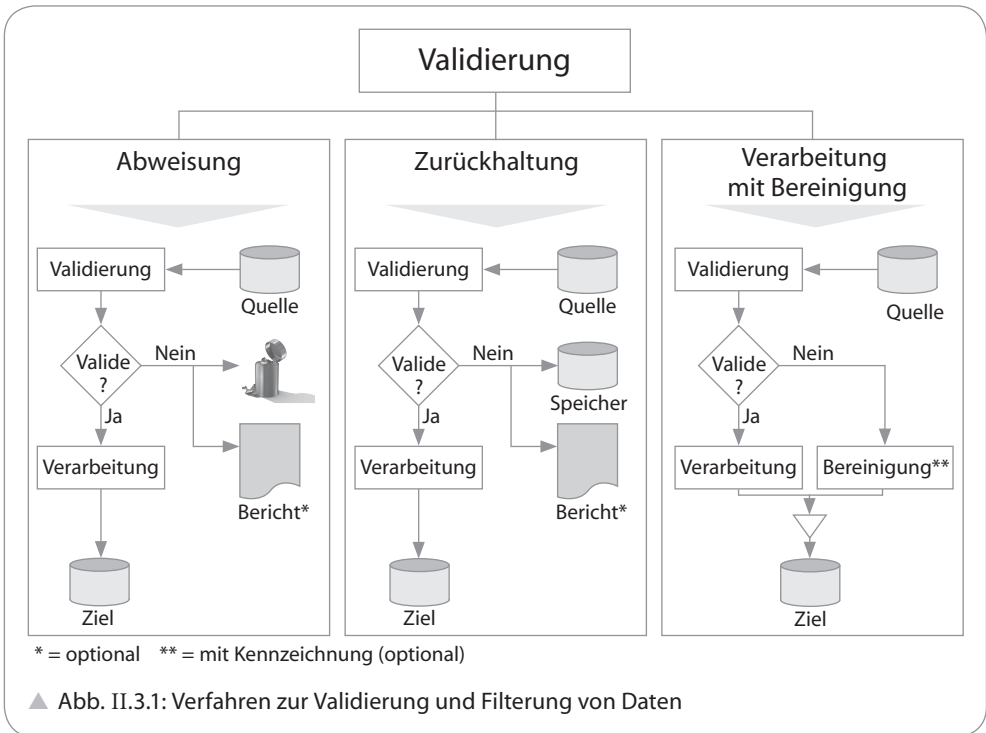
Auf **Datensatz-Ebene** überprüft die Validierung eine Struktur. Eine typische Prüfung: Ist der Wert für das Attribut `GESAMT_PREIS` das Produkt der Werte der Attribute `EINZEL_PREIS` und `MENGE`? Weitere Beispiele enthält Abschnitt II.2.3.2.

Auf **Datenobjekt-Ebene** werden Regeln für eine Sammlung von gleichartigen Strukturen (z. B. einer Tabelle) überprüft. Eine mögliche Prüffregel: Ist das Geburtsdatum aller Kunden in der Tabelle `KUNDEN` zu 98 Prozent mit einem gültigen Datum gefüllt? In Abschnitt II.2.3.3 sind weitere Beispiele beschrieben.

Eine Validierung auf **Bestands-Ebene** bezieht sich auf den Gesamt- (z. B. ein gesamtes Quellsystem) oder einen definierten Teil-Bestand der Daten (z. B. eine spezielle Ladedatei). Ein Beispiel: Haben alle Organisationseinheiten jeweils eine Datei mit den Verkaufszahlen für den aktuellen Monat angeliefert?

II.3.2 Filterung fehlerhafter Daten

Hat die Validierung fehlerhafte Daten erkannt, so werden diese ausgefiltert und gesondert behandelt. In der Praxis sind drei Verfahren mit jeweils zwei Varianten gebräuchlich: Abweisung, Zurückhaltung, Verarbeitung (siehe Abbildung II.3.1).



II.3.2.1 Abweisung der fehlerhaften Daten

Dieses sehr einfache Verfahren weist nicht verwendbare Daten ab, verarbeitet sie nicht weiter und protokolliert die Abweisung auch nicht. Werden einzelne Attributwerte abgewiesen, so werden diese in der Regel auf NULL gesetzt und anschließend mit dem validierten Rest geladen. Fehlerhafte Datensätze, Datenobjekte und Datenbestände werden komplett abgewiesen. In der Praxis kommt dieses Verfahren selten zum Einsatz, weil es gravierende Nachteile hat:

- › **Inkonsistenzen zwischen Quelle und Ziel:** Die fehlenden Daten sind nach der Abweisung noch an der Quelle (z. B. in der CRM⁹-Anwendung) vorhanden, fehlen aber in der BI-Anwendung. Vergleicht man diese Daten, kommt es zu Inkonsistenzen.
- › **Fehlender Nachweis:** Die abgewiesenen Daten werden nicht protokolliert. Das widerspricht gegebenenfalls vorgegebenen Compliance-Anforderungen, außerdem kann der Verbleib der fehlenden Daten im Nachgang nicht nachvollzogen werden.
- › **Fehlende Information:** Die Benutzer der nicht abgewiesenen Daten wissen nicht, wie viele und welche Daten abgewiesen wurden. Die Datenqualität der verbliebenen Daten lässt sich somit nicht quantifizieren – die Nutzbarkeit der Daten sinkt.
- › **Probleme bei Deltaladungen:** Werden nur die geänderten Daten in einem Deltaverfahren verarbeitet, werden diese bei einer erfolgten Abweisung verfälscht. Wurde der initiale Datensatz abgewiesen, fehlt dieser bei den nachfolgenden Aktualisierungen – diese können dann ebenfalls nicht verarbeitet werden. Wird eine Aktualisierung oder Stornierung abgewiesen, entspricht der Datenbestand nicht mehr der Realität.

⁹ CRM: Customer Relationship Management.

- › **Verminderte Datenqualität:** Selbst kleine, schnell und einfach behebbare Datenfehler führen zur Abweisung. In der Praxis ist die Anzahl der abgewiesenen Daten so hoch, dass die Vollständigkeit abnimmt und damit die Datenqualität stark sinkt.
- › **Fehlende Korrekturmöglichkeiten:** Die ausgefilterten Daten können in der BI-Anwendung nicht korrigiert und nachgeladen werden. Die einzige Möglichkeit zur Korrektur existiert an der Quelle (siehe Kapitel II.1), anschließend werden die korrigierten Daten erneut in die BI-Anwendung geladen.
- › **Großer Zeitverzug:** Durch die Abweisung der fehlerhaften Daten, deren Korrektur an der Quelle und dem Nachladen dauert es sehr lange, bis auch diese Daten im Ziel verfügbar sind. Stehen die Daten an der Quelle nicht mehr zur Verfügung, müssen diese zuerst aufwendig erneut erzeugt werden.

II.3.2.2 Abweisung der fehlerhaften Daten mit Bericht

Im Gegensatz zur reinen Abweisung erfasst man bei diesem Verfahren die abgewiesenen Daten in einem Bericht. Die fehlerhaften Daten selbst werden nicht gespeichert. Das Verfahren hat die gleichen Nachteile wie die Abweisung. Allerdings gibt es einen gedruckten oder elektronischen Nachweis für die abgewiesenen Daten.

In der Praxis zeigt sich leider häufig, dass die Adressaten (die Datenlieferanten an der Quelle) diese Berichte nicht lesen. Die Folge: Die fehlerhaften Daten werden nicht korrigiert, sodass diese in der Regel niemals in die BI-Anwendung geladen werden. Entweder man spart sich den Aufwand für deren Erstellung von Anfang an oder man regelt, wer die Berichtsinformationen umsetzt und die notwendigen Korrekturen durchführt.

II.3.2.3 Zurückhaltung der fehlerhaften Daten

Eine weitere Möglichkeit: Man speichert die fehlerhaften Daten in einem gesonderten Bereich. Dort stehen sie für nachfolgende Fehleranalysen und eine manuelle Weiterverarbeitung und Korrektur zur Verfügung. Dieses Verfahren hat weniger Nachteile als das Abweisen: Zwar bleiben die Inkonsistenzen zwischen Quelle und Ziel und die fehlende Information der Nutzer bestehen, dafür werden die fehlerhaften Daten eindeutig nachgewiesen und können (wenn auch nur manuell) korrigiert werden. Die Probleme mit den Deltaladeläufen, der verminderten Datenqualität und dem Zeitverzug werden abgemildert, da die fehlerhaften Daten im Nachgang zumindest teilweise korrigiert und nachgeladen werden. Vollständig lassen sich diese Probleme aber nicht lösen, da es aufgrund der Beziehungen zwischen den fehlerhaften und den validierten Daten sehr schwierig ist, diese getrennt voneinander zu korrigieren. Dadurch ist diese Korrektur zumeist unvollständig – und ein nennenswerter Rest fehlerbehafteter Daten bleibt.

Darüber hinaus ist mit diesem Verfahren ein schwerwiegender Nachteil verbunden, nämlich ein **hoher Ressourcenbedarf**: Das Speichern der fehlerhaften Daten verbraucht Speicherplatz, der im Laufe der Zeit kontinuierlich zunimmt. Selbst wenn ein Teil der Daten korrigiert, weiterverarbeitet und dann hier gelöscht wird, verbleibt ein nicht unerheblicher Restbestand. Diese erhöhte Datenmenge verlangsamt auch die Lade- und Zugriffszeiten und kann daher für das gesamte Data Warehouse problematisch sein. Um diese Problematik zu minimieren, sollte man folgende Punkte beachten:

- › Jeden Datensatz nur einmal abspeichern, auch wenn er mehrfach in mehreren Validierungsläufen zurückgehalten wurde.
- › Nicht für jede nicht erfüllte Validierungsregel den Datensatz duplizieren. Soll die Ursache für das Zurückhalten im Datensatz dokumentiert werden, so werden in **einem** Datensatz **alle** verletzten Validierungsregeln gemeinsam angehängt.
- › Nach der Aufbewahrungsfrist Altdaten regelmäßig archivieren oder löschen.
- › Ausreichend Platz für die Daten bereitstellen, dabei das Wachstum berücksichtigen.

II.3.2.4 Zurückhaltung der fehlerhaften Daten mit Bericht

Bei diesem Verfahren wird zusätzlich ein gedruckter oder elektronischer Bericht erzeugt. Dieser enthält eine Übersicht über die zurückgehaltenen Daten und erleichtert somit die Weiterverarbeitung und die Korrektur. Gleichzeitig kann man den Bericht für das Berichtswesen verwenden. Die Nachteile für das Zurückhalten der Daten gelten auch hier. Außerdem lehrt die Praxis, dass nur wenige Benutzer diesen Bericht auch lesen (siehe Abschnitt II.3.2.2).

II.3.2.5 Verarbeitung der fehlerhaften Daten mit Bereinigung

In diesem Falle werden die fehlerhaften Daten weder abgewiesen noch zurückgehalten, sondern in der BI-Anwendung weiterverarbeitet. Dort werden sie automatisch bereinigt. Fehler, die sich nicht automatisch beheben lassen, werden einem Sachbearbeiter zugeführt, der diese dann manuell bearbeitet. In der Praxis ist dieses Verfahren sehr gebräuchlich.

Die Nachteile dieses Verfahrens sind:

- › **Fehleranfälligkeit:** Durch die automatisierte Bereinigung können neue Datenfehler entstehen, welche die Datenqualität vermindern statt erhöhen.
- › **Komplexität, Laufzeit:** Die Bereinigung ist in der Praxis sehr komplex und laufzeitintensiv. Schnell beeinträchtigt sie die reguläre Verarbeitung der validen Daten und verlangsamt die Verarbeitungsprozesse insgesamt.
- › **Fehlende Information:** Die bereinigten Daten lassen sich nicht von den nicht bereinigten unterscheiden, sodass die korrigierten Werte für den Benutzer nicht eindeutig als solche gekennzeichnet werden können.

Mehr Informationen zur Bereinigung von Daten enthält Kapitel II.4.

II.3.2.6 Verarbeitung der fehlerhaften Daten mit Bereinigung und Kennzeichnung

Während der Verarbeitung mit Bereinigung kennzeichnet die Anwendung die fehlerhaften Daten. Dabei ist es sogar möglich, einen Wert für die Qualität an die jeweiligen Datenelemente anzufügen. Diese zusätzlichen Informationen lassen sich vielfältig verwenden:

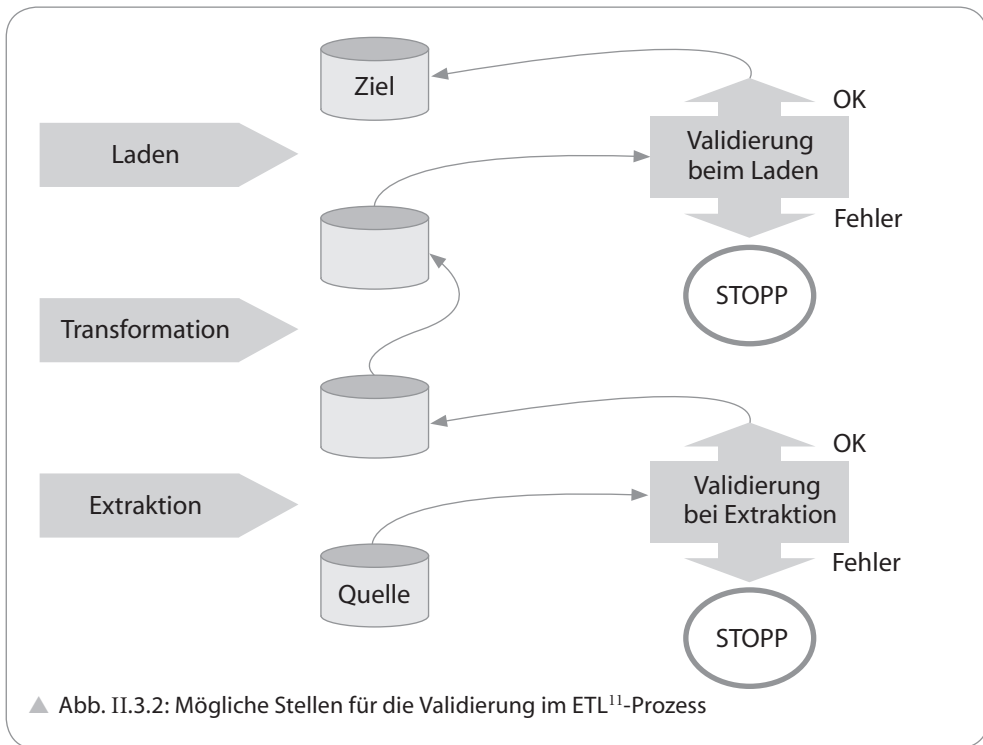
- › Der Benutzer kann die fehlerhaften Daten direkt und eindeutig identifizieren und über ihre Verwendung selbstständig entscheiden.
- › Die nachfolgenden Verarbeitungsprozesse können abhängig von der Qualität der Fehler individuell gesteuert werden (z. B. Daten mit einer Qualität unter einem Grenzwert werden nicht weiterverarbeitet).
- › Es können sehr einfach Berichte über Art und Umfang der Fehler erstellt werden.

- › Wurden die fehlerhaften Daten nach dem Laden auch an der Quelle korrigiert, lassen sich die zugehörigen, in der BI-Anwendung korrigierten Daten sehr leicht zuordnen und durch die an der Quelle korrigierten Daten ersetzen.

Der Nachteil: Das Verfahren ist sehr komplex, der Aufwand dementsprechend hoch. Obwohl das Verfahren die besten Ergebnisse liefert, kommt es in der Praxis selten zum Einsatz.

II.3.3 Validierung bei Extraktion oder Laden

An welcher Stelle im Datenladeprozess sollen die Daten validiert und gefiltert werden? Grundsätzlich sind zwei Stellen möglich: bei der Extraktion und/oder beim Laden der Daten (siehe Abbildung II.3.2). In beiden Fällen ist zu verhindern, dass bei der Validierung als fehlerhaft erkannte Daten unkorrigiert in die Datenbereiche des Data Warehouse (z. B. Data Marts, Core Data Warehouse, ODS¹⁰-Objekte) gelangen.



II.3.3.1 Validierung bei der Extraktion

Die Daten werden direkt bei der Extraktion in den zugehörigen Prozessen aus den Quellen validiert, sodass fehlerhafte Daten frühzeitig erkannt und ausgefiltert werden. Da in den meisten Projekten die fehlerhaften Daten abgewiesen werden (siehe Abschnitt II.3.2.1), erreichen sie die BI-Anwendung nicht und können deren Qualität nicht beeinträchtigen. Allerdings bleiben die Nachteile einer Abweisung bestehen. Wird bei der Extraktion validiert, kommen

¹⁰ ODS: Operational Data Store.

¹¹ ETL: Extraktion, Transformation, Laden.

die komplexeren Verfahren (siehe Abbildung II.3.1) selten zum Einsatz. Schließlich sind die verfügbaren Möglichkeiten bei der Extraktion sehr eingeschränkt. So lassen sich zum Beispiel Inkonsistenzen in der Regel erst durch einen Vergleich mit den in anderen Prozessen extrahierten oder den historischen Daten aus der BI-Anwendung validieren.

In der Praxis kommen bei der Extraktion nur vergleichsweise einfache Validierungsarten wie Datentyp-, Domain- und einfache Integritäts-Validierungen zum Einsatz. Diese erkennen aber nur einen kleinen Teil der fehlerhaften Daten, da z. B. Transaktionsdatensätze von Natur aus (fast) immer integer sind und dadurch Integritäts-Validierungen hierin kaum Fehler finden können. Dafür sind die Verfahren sehr einfach und der damit verbundene Aufwand ist vergleichsweise gering.

II.3.3.2 Validierung beim Laden

Werden die Daten beim Laden in die Ziele (z. B. Core Data Warehouse, Data Marts) validiert, befinden sie sich bereits in der BI-Anwendung und sind schon transformiert. Eingesetzt wird diese Validierung insbesondere beim Laden in diejenigen abhängigen Data Marts, die spezifische Regeln für die Validierung besitzen. Hier kann die Validierung nicht vorher erfolgen, da diese spezifischen Regeln nicht für das übergreifend genutzte Data Warehouse und andere Data Marts gelten.

Aufgrund der während der Transformation durchgeführten Standardisierung (siehe Kapitel II.4) lassen sich bei dieser Validierung vorhandene Duplikate besser und eindeutiger erkennen. Da alle geladenen und historisierten Daten aus dem Data Warehouse (inklusive Data Marts) verwendet werden, sind auch umfassendere und komplexere Validierungen möglich. Auf der anderen Seite steigt die Gefahr, mit der Validierung unbeabsichtigt an sich korrekte Daten auszufiltern und die Datenqualität zu verringern anstatt zu erhöhen. Dieser Gefahr muss man mit einem ständigen Data Quality Monitoring (siehe Kapitel II.8) entgegenwirken.

Möchte man diesen Nachteil minimieren, so kann man eine Kopie der benötigten Tabellen inklusive Daten aus der BI-Anwendung erstellen. Die neuen Daten werden nach der Transformation zunächst in diese Kopie gespeichert. Dadurch ist es möglich, genauso viele Fehler zu finden wie beim direkten Laden in das Ziel. Da es sich um eine Kopie handelt, beeinträchtigen die Fehler die Qualität des originären Zieles nicht. In das richtige Ziel gespeichert werden nur die nach der Validierung als fehlerfrei identifizierten Daten. Die fehlerhaften Daten lassen sich mit diesem Wissen anschließend besser bereinigen. Diese Vorgehensweise bietet die umfassendsten Validierungs- und Korrektur-Möglichkeiten. Leider sind die Kosten für die Implementierung höher, die Laufzeiten der ETL-Prozesse länger und der Ressourcenverbrauch durch die doppelte Speicherung größer. Deshalb ist dieses Verfahren in der Praxis sehr selten.

II.3.3.3 Validierung bei der Extraktion und beim Laden

Egal ob Validierung bei der Extraktion oder beim Laden: Beide Verfahren haben ihre Vor- und Nachteile und sind je nach Problem besser oder schlechter geeignet. Aus diesen Gründen empfiehlt sich ein kombinierter, auf die jeweiligen Anforderungen ausgerichteter Ansatz: Validierungen, die bei der Extraktion vorteilhaft und möglich sind, werden direkt am Anfang bei der Extraktion durchgeführt. Validierungen, welche sich erst nach der Transformation

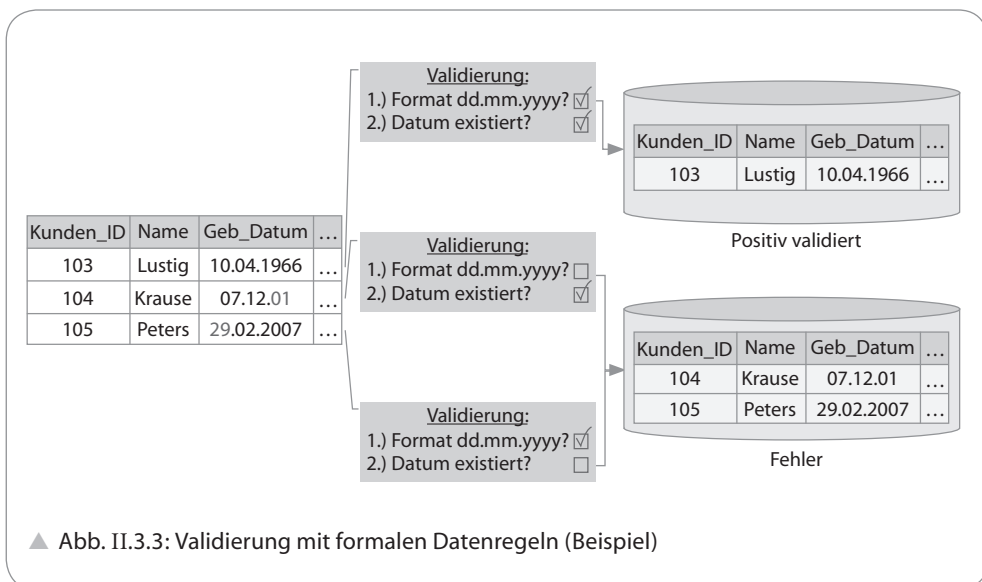
erfolgreich durchführen lassen, erfolgen erst beim Laden. Damit verringern sich auch die Laufzeit und der Ressourcenbedarf des ETL-Prozesses, weil die schon bei der Extraktion als fehlerhaft identifizierten und anschließend aussortierten Daten nicht erst durch den nachfolgenden Transformations- und Ladeprozess mitgeführt und verarbeitet werden müssen.

II.3.4 Arten der Datenvalidierung

Nun stellt sich noch die Frage: Wie und womit wird die Validierung durchgeführt?

Die besten Ergebnisse liefert in der Regel eine Validierung, die formale Daten- und fachliche Geschäftsregeln (siehe auch Kapitel II.2) kombiniert. Die formalen Datenregeln werden beim Data Profiling aus der Analyse erstellt. Beispiele sind die abgeleiteten Datenregeln aus der Datentyp- oder Domänen-Analyse in Kapitel II.4. Die Geschäftsregeln resultieren aus dem fachlichen Wissen der Experten und nutzen dieses für eine inhaltliche Validierung.

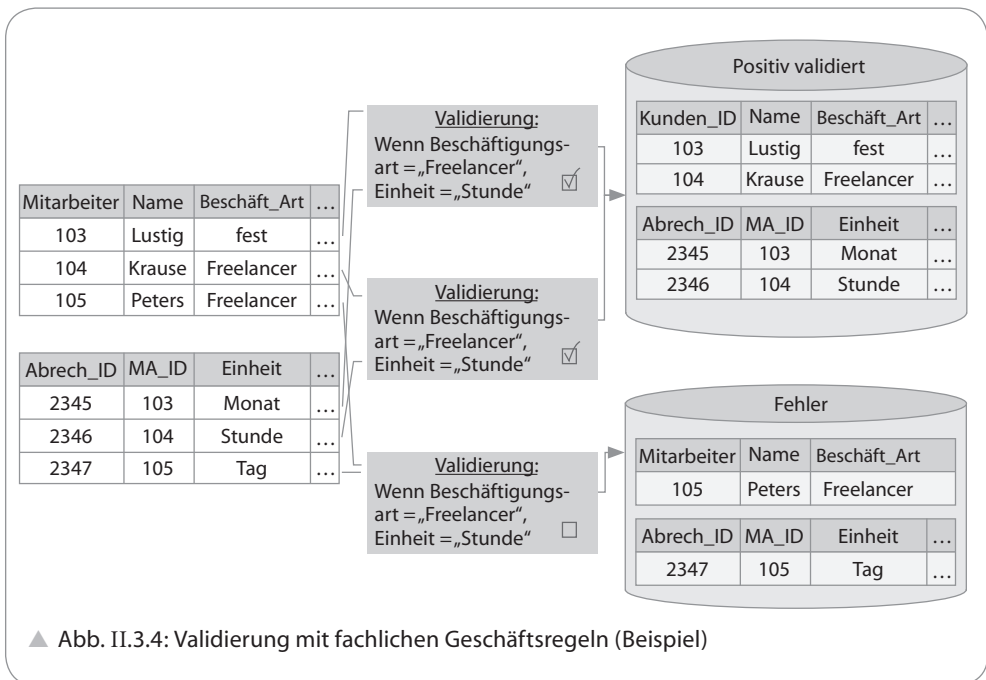
Die **formalen Datenregeln** setzt man in der Praxis überwiegend zur Validierung bei der Extraktion ein, da diese relativ einfach zu realisieren sind und auch allein auf einzelne Attribute und Datensätze angewendet werden können. Sie benötigen für die Validierung keinen Zugriff auf andere Datenobjekte oder historisierte Daten aus der BI-Anwendung. In dem Beispiel aus Abbildung II.3.3 wird das Attribut GEB_DATUM (= Geburtsdatum) mit Hilfe von zwei formalen Datenregeln geprüft: Liegen die Werte in dem geforderten Format vor? Und handelt es sich um ein gültiges, real existierendes Datum? Für den Kunden mit der ID 103 sind beide Regeln erfüllt, die Validierung endet mit einem positiven Ergebnis. Für die Kunden mit den IDs 104 und 105 wird jeweils eine der Regeln verletzt, die Validierung endet mit einem Fehler.



Die Validierung mit **fachlichen Geschäftsregeln** findet bei der Extraktion statt, aber hauptsächlich beim Laden. Damit können Daten auch mit komplexen Regeln validiert werden, wozu auch historische Daten aus der BI-Anwendung oder andere Datenobjekte verwendet

werden können. Beispiele zu fachlichen Geschäftsregeln sind ebenfalls in Kapitel II.2 dargestellt, z. B. in den „Regeln zu Wertebereichen“ in Abschnitt II.2.3.1.2 oder den „Regeln für Zeitreihen“ in Abschnitt II.2.3.3.2. In dem Beispiel aus Abbildung II.3.4 lautet die fachliche Geschäftsregel: „Freelancer werden ausschließlich nach Stunden bezahlt.“ Das Attribut `BESCHÄFT_ART` enthält die Information, ob ein Mitarbeiter Freelancer ist; das Attribut `EINHEIT` gibt die Einheit für die Bezahlung an. Die Validierung für die Mitarbeiter 103 und 104 ist erfolgreich, da der Mitarbeiter 103 ein Festangestellter ist und die Abrechnungseinheit für den Mitarbeiter 104 (Freelancer) tatsächlich *Stunde* ist. Die Validierung für den Mitarbeiter 105 meldet folgerichtig einen Fehler, da die Beschäftigungsart *Freelancer* und die Einheit nicht *Stunde* ist.

Hinweis: Die Validierung allein kann nicht entscheiden, welcher der beiden Einträge (*Freelancer*, *Stunde*) fehlerhaft ist. Sie erkennt nur, dass die zugehörige Geschäftsregel verletzt wurde.

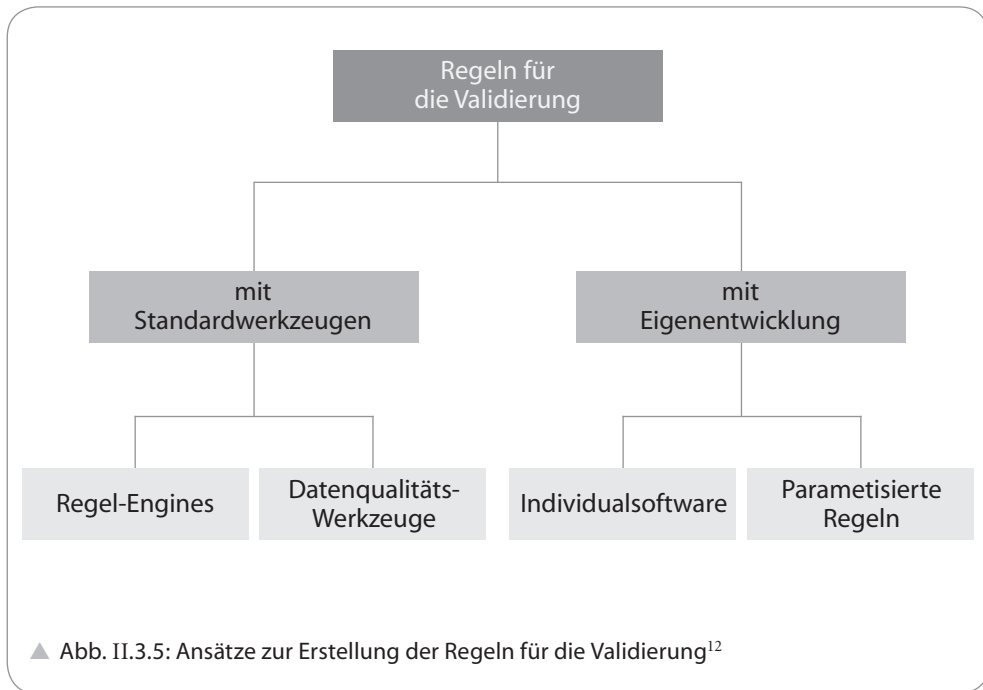


Möglich ist ebenfalls eine Validierung mit **Referenzdaten**. Auch sie kann sowohl beim Laden als auch bei der Extraktion eingesetzt werden. Referenzdaten sind vertrauenswürdige Daten mit hoher Reputation, sie können aus internen oder externen Quellen stammen. Ein Beispiel sind Adressdaten, die von externen Anbietern wie der Deutschen Post bezogen werden. Die vorhandenen Daten werden bei der Validierung mit diesen Referenzdaten verglichen, wobei die Referenzdaten den Master darstellen. Abweichungen werden als Fehler interpretiert, anschließend werden üblicherweise die Werte mit Hilfe der Referenzdaten korrigiert.

Die nachfolgende Bereinigung der fehlerhaften Daten ist Thema in Kapitel II.4.

II.3.5 Erstellung der Validierungsregeln und Speicherung der Ergebnisse

Nun ist noch zu klären, womit die Validierungsregeln erstellt und wo die Ergebnisse gespeichert werden sollen. Grundsätzlich kommen für die Erstellung der Regeln vier unterschiedliche Ansätze (siehe Abbildung II.3.5) in Frage.



Einerseits erstellen und validieren Standardwerkzeuge diese Regeln. In Frage kommen entweder spezielle Engines für Daten- und Geschäftsregeln oder aber Datenqualitäts-Werkzeugsuiten, die teilweise diese Funktionalität enthalten. Allerdings sind die Möglichkeiten dieser Standardwerkzeuge noch eingeschränkt, sodass nicht alle Regeln mit ihrer Hilfe definiert werden können. Außerdem sind sie teilweise schwer zu bedienen und decken nicht alle Anforderungen (z. B. Gruppierung von Regeln, Bildung von Fehlerklassen) ab. Allerdings entwickeln sich diese Werkzeuge rasch weiter. Ein starker Trend zu kompletten, leistungsstarken Datenqualitäts-Suiten ist erkennbar, weshalb in nächster Zeit mit verwendbaren Lösungen der Anbieter zu rechnen ist. Da diese Suiten in der Regel auch ETL-, Monitoring- und Berichts-Funktionalitäten anbieten, die eine vollständige Integration und Wiederverwendung der Regeln ermöglichen, sollte man den aktuellen Einsatzstand und die Verwendung der Suiten vor Projektbeginn überprüfen.

Andererseits können Projektteams diese Regeln individuell erstellen und validieren. Entweder entwickeln sie eigenständige Programme (z. B. SQL-Skripte) oder parametrisierbare Regel-Engines. Diese Individualentwicklungen decken alle speziellen Anforderungen ab, doch bleiben die Nachteile für Individualsoftware: der hohe Aufwand und die schlechte Wieder-

¹² Vgl. Maydanchik 2008.

verwendbarkeit in anderen Projekten. Außerdem muss man zusätzliche Anforderungen (wie z. B. Integration in ETL-Prozesse für das Monitoring) selbst realisieren – sofern die verwendeten Werkzeuge überhaupt offene Schnittstellen für diese Integration zur Verfügung stellen.

Es bietet sich an, diese Regeln und die Ergebnisse der Validierungen in einem eigenen Repository zu speichern. Hat man kommerzielle Werkzeuge im Einsatz, ist man leider meistens auf die vom Hersteller gewählte Lösung angewiesen. Sieht diese keine Speicherung in einem Repository vor, kann eine solche nur mit hohem Aufwand individuell realisiert werden. In der Praxis sind damit häufig Funktionseinschränkungen und eine verminderte Bedienbarkeit verbunden.

Weiter sollte man darauf achten, dass andere Werkzeuge das Repository ebenfalls benutzen. Ansonsten können weder diese Regeln z. B. im ETL-Prozess für die Fehlererkennung und -bereinigung (siehe Kapitel II.4) noch die Validierungsergebnisse für das Monitoring (siehe Kapitel II.8) verwendet werden.

II.3.6 Empfehlungen

Es gibt keine generelle Antwort auf die Frage, welches Verfahren an welcher Stelle in welcher Art bei der Validierung in einem Projekt eingesetzt werden soll. Genauso wenig gibt es eine überall verwendbare „Musterlösung“. Alle möglichen Kombinationen sind denkbar und können je nach Aufgabe auch das beste Ergebnis liefern. Erfahrene Datenqualitäts-Experten können mit den vielfältigen Auswahlmöglichkeiten eine maßgeschneiderte Lösung entwerfen. Damit das funktioniert, müssen sich diese Experten **vor** der Durchführung Gedanken über die Aufgaben des speziellen Projektes machen, die richtige und zweckmäßigste Kombination unter den gegebenen Randbedingungen (z. B. Projektlaufzeit, Budget, Gefährdungspotenzial) identifizieren, diese planen und **dann** realisieren lassen. Das Projektteam sollte sich aber immer die Nachteile der gewählten Lösung vor Augen halten und den damit verbundenen Risiken aktiv entgegenwirken.