ETL mit PL/SQL – Erfahrungen aus der Praxis

Alexander Mendle
SHS VIVEON AG
München

Schlüsselworte:

ETL, PL/SQL, DWH, Metadaten, Ladelaufsteuerung, Logging, Datenqualität

Einleitung

PL/SQL findet als Erweiterung von Datenbanken um eine Programmiermöglichkeit weite Verbreitung. Durch die breite Verfügbarkeit von PL/SQL ist es möglich, Programmcode direkt auf der Datenbank auszuführen, sowie SQL Code direkt in das Programm mit einzubauen.
Dies sind ideale Voraussetzungen für den Einsatz bei der Programmierung von ETL-Strecken. Im Vergleich zu verfügbaren Werkzeugen fällt das Ergebnis von ETLs mit PL/SQL qualitativ äußerst heterogen aus. Der Ersteller von ETL Routinen hat alle erdenklichen Freiheiten, weshalb das Ergebnis stark vom Know-How des Entwicklers und der entsprechenden Weitsicht in der Entwicklung abhängt.
Dieser Vortrag soll Möglichkeiten und Kniffe aufzeigen, wie professionelle ETL-Strecken mit PL/SQL umgesetzt werden können, um das Beste aus beiden Welten mit der Flexibilität von PL/SQL zu verbinden.

ETL mit Tools und PL/SQL – einige Unterschiede

In der Entwicklung von ETL Strecken zeigen sich einige Unterschiede zwischen der Verwendung eines ETL Tools und der Umsetzung mit PL/SQL. Einige dieser Unterschiede werden im Folgenden kurz beleuchtet, bevor schließlich Wege zum professionellen Entwickeln von ETLs aufgezeigt werden.

Spezifika PL/SQL

Die Verwendung von PL/SQL bietet zweifelsohne einen sehr hohen Grad an Individualisierbarkeit. Jede denkbare Funktion zur Datenmanipulation, sofern sie von der Datenbank unterstützt wird, kann in beliebiger Komplexität umgesetzt werden. Dabei bieten derart umgesetzte Funktionen auch einen hohen Grad an Zukunftsfähigkeit, da bei Releasewechseln und dergleichen keine Abhängigkeiten zwischen Tool und Datenbank beachtet werden müssen. Ressourcen für die Entwicklung und Wartung von PL/SQL Programmen stehen am Markt in großer Zahl zur Verfügung, nicht zuletzt deshalb da PL/SQL nicht auf den DWH und BI Markt fokussiert ist, wie der Großteil an ETL Tools. Das Tuning von ETL-Strecken auf Basis von PL/SQL stellt sich sehr transparent dar, da direkt auf der Datenbank gearbeitet werden kann und dazwischenliegende Applikationsschichten nicht beachtet werden müssen.
Ein wesentlicher Punkt im flexiblen Einsatz von PL/SQL zu ETL-Zwecken besteht in der Möglichkeit, Code-Generatoren zu verwenden, um z. B. metadatengetrieben Datenstrukturen aufzubauen. Im Gegensatz dazu bieten doch auch ETL Tools einige Eigenheiten.

Spezifika ETL mit Tool

Ein wesentlicher Unterschied im Vergleich zur Gestaltung von ETLs mit PL/SQL ist, dass ein ETL Tool stets Metadaten vorhält. Damit sind unter anderem in aller Regel folgende Informationen zu einem ETL vorhanden:

- Systematisches Logging zur Unterstützung von Betrieb und Entwicklung
- Zentrale Ablage von Mappings, Datenflüssen, Tabellen- und Spaltendefinitionen
- Informationen zu Ladeläufen

Diese Daten sind auch Grundlage für einige Funktionalitäten, welche ETL Tools gemeinhin bereitstellen. Bei der Verwendung von PL/SQL müssen diese Funktionalitäten im benötigten Umfang zur Verfügung gestellt werden, insofern hier nicht immer das Rad neu erfunden soll. Folgende Möglichkeiten bieten ETL Tools in aller Regel an:

- Breite Unterstützung bei einer Vielzahl von Extraktions-, Transformations- und Datenmanipulationsoperationen.
- Standardverfahren zur Überwachung der Datenqualität
- Einheitliches und systematisches Logging
- Unterstützung bei der Ablaufsteuerung, beispielsweise durch Verwaltung von Abhängigkeiten
- Versionsverwaltung und Historisierung, teils mit Releasemanagment und Unterstützung beim Deployment
- Funktionen zur Steigerung der Produktivität
    - Wiederverwendung von Bausteinen
    - Hilfen bei Erkennung von Tabellenstrukturen und Abhängigkeiten
    - Unterstützung in der Entwicklung mit Debugging und Breakpoints

Neben diesen offensichtlichen Unterschieden zwischen ETL-Entwicklung mit Tool und mit PL/SQL gibt es noch eine Reihe weiterer Faktoren, welche die beiden Varianten unterscheiden. Die Lizenzkosten sind für ein Tool im Allgemeinen höher, da zusätzlich zu den Datenbanklizenzen noch Lizenzen für das ETL-Tool angeschafft werden müssen. Im Falle von Migrationen, Releasewechseln und ähnlichen Umzugsplänen sind stets zusätzliche Abhängigkeiten im Zusammenspiel zwischen Tool und Datenbank zu beachten. Diese Abhängigkeit spielt auch beim Performance-Tuning eine entscheidende Rolle. Nur mit profunder Kenntnis beider Technologien können hier Vorteile erzielt werden. In der Flexibilität der Anwendung sind dem Tool jedoch Grenzen gesetzt. Eine Erweiterung über die enthaltenen Funktionen hinaus erfordert entweder Erweiterungen unter Einsatz von sehr speziellem Know-How, oder die Erweiterung mit Funktionalität auf der Datenbankseite, wie z. B. mit PL/SQL.
Mit dieser kurzen Darstellung der Spezifika von ETL mit Tool und mit PL/SQL sind nun einige Gedanken aufgeworfen worden, die im Folgenden aus einer anderen Perspektive heraus beleuchtet werden.

Professionelles ETL mit PL/SQL

Die folgenden Ausführungen beschäftigen sich nun damit, in welchen Bereichen Maßnahmen ergriffen werden können, um die oben genannten Vorzüge der beiden Wege bei der Entwicklung von ETLs mit PL/SQL zu genießen. Dazu wird das Kapitel in vier Abschnitte geteilt. Zwei davon, die Teile „Metadaten & Bibliothek“ sowie der Teil „Hilfsmittel“ beschäftigen sich mit Datenbanken, Dateien, Programmen und anderen Software-Artefakten in Verbindung mit ETL-Prozessen, während sich die beiden Teile „Organisation“ und „Prozesse“ mit der Gestaltung der Arbeit an ETL-Prozessen beschäftigen.

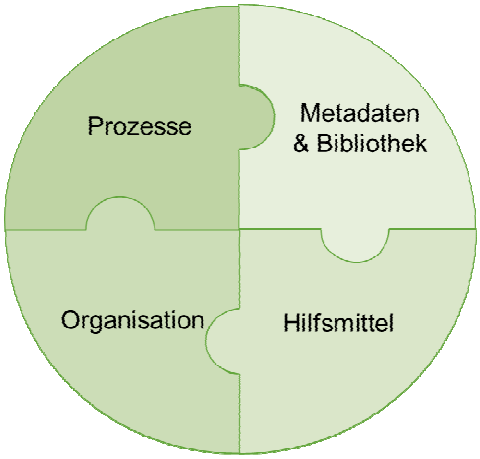


Abb. 1: Einteilung der weiteren Erläuterungen in vier Bereiche

Metadaten und Bibliothek

Dieses Kapitel steht unter folgender These: Metadaten gibt es, wenn ETL-spezifische Funktionen und Informationen in bekannter, gemeinsamer und erweiterbarer Struktur bereitgestellt werden. Grundlage dieser These ist die Überlegung, dass zur Realisierung von Software, welche eine bestimmte Funktionalität hat, auch Datenspeicher gehören. Wird nun weiterhin angenommen, dass Software für die Gestaltung von ETL-Prozessen spezifische Funktionen enthält, so enthält sie damit auch die entsprechenden Datenspeicher. Bei geschickter und praktikabler Gestaltung von Funktionalität und Datenspeicher, so die Folgerung, sollte es nun möglich sein eine notwendige Basis für Metadaten zu schaffen, die gemeinsam nutzbar und erweiterbar ist. Folgende Metainformationen sind Beispiele für vernetzbare Datenstrukturen, die mit konkreten Funktionalitäten von ETL Prozessen gekoppelt sind.

- Datenstrukturen von Quelle und Senke
- Transformationsregeln
- Qualitätsinformationen
- Abhängigkeiten
- Ladelaufinformationen und Fehler (Log)

Um die Daten zentral verfügbar zu halten und starke Fragmentierung zu vermeiden wird empfohlen, eine gemeinsame Bibliothek bereitzustellen, welche diese Daten und bestimmte ETL-Funktionalitäten integriert. Diese Bibliothek steht zentral allen Entwicklern zur Verfügung, und enthält Funktionalitäten z. B. für

- Logging
- Fehlerbehandlung (sowohl Datenfehler als auch Ablauffehler)
- Ladelaufkontrolle und Wiederanlauf
- Fertige Bausteine für
    - Datumsfunktionen (Feiertage, Betriebskalender)
    - Erstellung und Verwaltung von Partitionen und Indizes
    - Ausführung von dynamischem SQL
    - Pivotierung
    - Lookups für Konstanten, unternehmensweite Informationen
    - u. a. nach Bedarf

Folgendes Code-Beispiel zeigt einen Aufruf einer Loggingfunktion während eines ETL Laufs. Die Logging-Funktionalität wird von einem zentralen Package „pkg_bib_logging“ bereitgestellt.

```
dwh_bib.pkg_bib_logging(ETL_ID,DATE,RUN_ID,DATASET_ID,ERRMSG,SQLERR);
```

Listing 1: Logging mit einer zentralen Funktion aus der Bibliothek (Beispiel).

Von besonderer Bedeutung ist dabei, dass diese Bibliothek keinen monolithischen Charakter haben sollte. Sie ist von vitaler Bedeutung für die Erzeugung und Aktualisierung von Metadaten. Nur bei Nutzung der Bibliothek stehen auch die entsprechenden Metadaten zur Verfügung. In diesem Sinne wird auch der Ausbau der gemeinsamen Bibliothek empfohlen. Die Planung sollte dabei von der gemeinsamen Datenstruktur ausgehen, und auf leicht verwendbare und produktiv einsetzbare Funktionen zielen.

Prozesse

Bei der Verwendung dieser gemeinsamen Bibliothek im Rahmen der ETL-Entwicklung, wie auch im Laufe des Ausbaus der Metadaten bleibt es nicht aus, dass Verbesserungsvorschläge und Änderungsanforderungen von Seiten der ETL-Entwickler entstehen. Ebenso sind im Entwicklungsprozess für ETLs an einigen Stellen spezifische Eigenschaften zu beachten. Bei der Betrachtung der Prozesse im Kontext ETL-Entwicklung wird auf folgende drei Bereiche zu eingegangen:

- Entwicklungs- und Betriebsprozesse
- Freigabeprozesse
- Planung der Weiterentwicklung von Metadaten & Bibliothek

Viele Teile solcher Prozesse werden aus einem Standardprozess für Softwareentwicklung übernommen. An einigen Stellen eines solchen Standardprozessen sollten bei der Anwendung im Bereich ETLs mit PL/SQL einige Hinweise berücksichtigt werden, damit auch dauerhaft hochwertige Software entstehen kann. Zur Illustration wir der im Folgenden abgebildete, generische Prozess zur Softwareentwicklung verwendet.

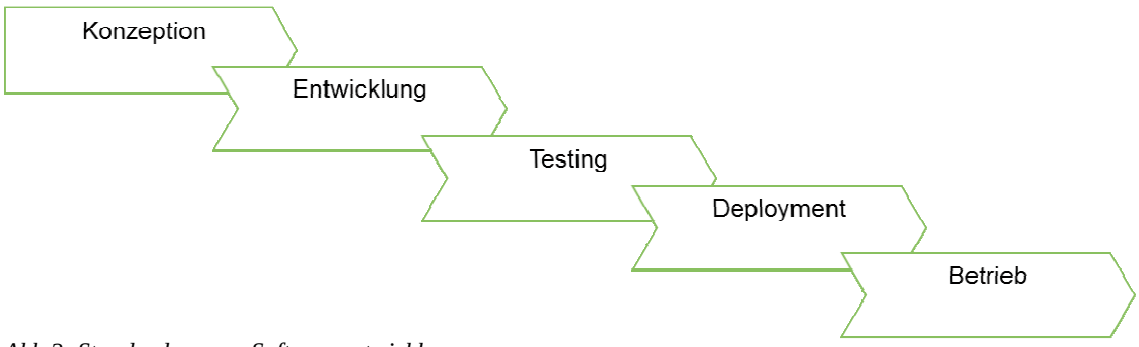


Abb.2: Standardprozess Softwareentwicklung

Ein solcher Prozess kann jedem Lehrbuch der Informatik entnommen werden. Egal ob die Entwicklung nach einem Wasserfallmodell wie hier oder in einem agilen Verfahren entwickelt wird: Da bei der Entwicklung von ETLs stets die bestehende Landschaft, aber insbesondere die Bibliothek wie auch die Metadaten von entscheidender Bedeutung für die Qualität der gesamten DWH Landschaft sind, sollten im Entwicklungsprozesse folgende Punkte besonders beachtet werden.
Bei der **Konzeption** ist sicherzustellen, dass unter den Entwicklern ausreichend Kenntnis über die gemeinsamen Bibliotheken, sowie die Metadaten herrschen. Zudem könnte die Erstellung einer Entwicklungsrichtlinie in Betracht gezogen werden, welche z. B. die Definition von Testfällen bereits in diesem Stadium der Entwicklung vorsehen könnte.
Während der **Entwicklungsphase** ist Sorge dafür zu tragen, dass die Entwicklung ausgehend von entsprechenden Templates erfolgt, und die Bibliotheken sowie die entsprechenden Repositorien und Versionierungssysteme genutzt werden. Damit sollte die Integration des neuen Projekts in die

bestehende Metadatenlandschaft gelingen. Diese Randbedingungen sollen auch beim **Testing** sowie bei ggf. notwendigen Abnahmen berücksichtigt werden.
Der Schritt in den Produktivbetrieb sollte einer besonderen Kontrolle unterliegen. Hier hat es sich als nützlich erwiesen, **Deployments** ausschließlich von in einem Versionierungssystem abgelegtem Code vorzunehmen. Zudem ist dies die Gelegenheit, den ETL Code im Hinblick auf neue Ideen für die Bibliothek zu untersuchen.

Hier knüpft ein neu zu gestaltender Prozess an. Die Bibliotheken sowie die Metadaten müssen ebenso betrieben und weiterentwickelt werden. Als Startpunkte diese Prozesses existieren zwei Prozessschritte, die sowohl einzeln wie auch zusammen einsetzbar sind: Die Aufnahme neuer Ideen für die Bibliothek, sowie ein Review mit wichtigen Nutzern.
Diese beiden Schritte sind Quellen für die Planung der Weiterentwicklung der Bibliothek. Steht eine gemeinsame Roadmap fest, kann die Weiterentwicklung gestartet werden und die entsprechend neuen Bestandteile in den Betrieb übernommen werden.

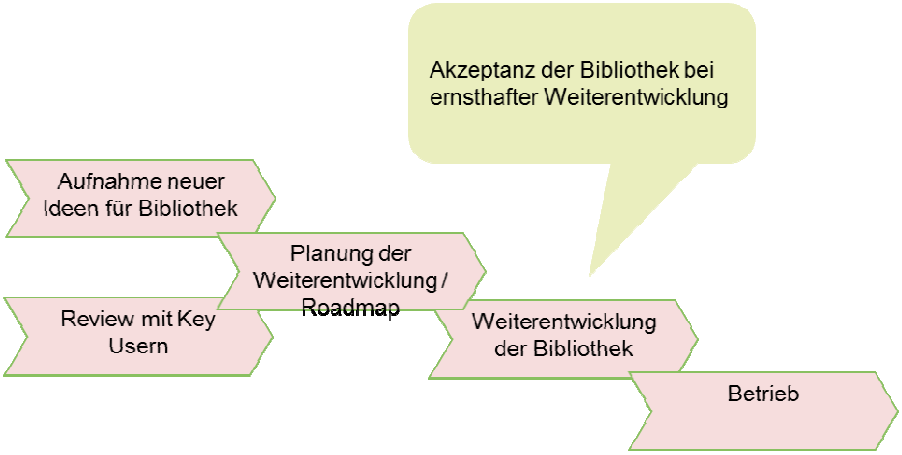


Abb.3: Weiterentwicklung der Bibliothek

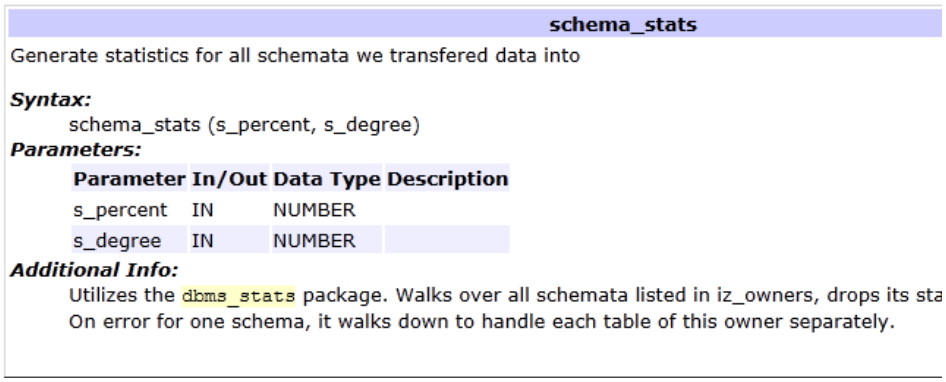
Nur durch kontinuierliche Weiterentwicklung der Bibliothek und der Metadatenbasis kann dauerhaft die Nutzungsintensität hoch gehalten und damit die Akzeptanz bei den Entwicklern sichergestellt werden.

Werkzeuge

Bei der Gestaltung von Prozessen und der Umsetzung von ETLs mit PL/SQL können eine Vielzahl von Werkzeugen wertvolle Unterstützungsarbeit leisten. Dabei ist es stark von der Umgebung abhängig ob ein Einsatz sinnvoll ist, ob er technisch möglich ist und natürlich ob die Notwendigkeit besteht. Solche Werkzeuge bieten sich z. B. für die Unterstützung in folgenden Fällen an:

- Automatisierte Erstellung von ETL-Dokumentation, hier ist z. B. die Software *HyperSQL* verfügbar (http://projects.izzysoft.de/trac/hypersql).
- Für die Versionsverwaltung kommt z. B. Subversion (SVN, http://subversion.apache.org/) in Frage.
- Mit einer Software zur Testautomatisierung wie z. B. FitNesse (http://fitnesse.org/) können vordefinierte Testfälle verwaltet und in Folge ohne weiteren Zusatzaufwand automatisch durchgeführt und protokolliert werden.

- Abnahmeprotokolle, pragmatisch mit einer Tabellenkalkulation erstellt, helfen dabei, vollumfängliche Prüfungen von PL/SQL Code zuverlässig und in wiederholbarer Qualität durchzuführen.



schema_stats

Generate statistics for all schemata we transfered data into

Syntax:

schema_stats (s_percent, s_degree)

Parameters:

Parameter	In/Out	Data Type	Description
s_percent	IN	NUMBER	
s_degree	IN	NUMBER	

Additional Info:

Utilizes the `dbms_stats` package. Walks over all schemata listed in iz_owners, drops its sta
On error for one schema, it walks down to handle each table of this owner separately.

Abb.4: Beispiel für eine automatisch erzeugte Dokumentation einer PL/SQL Prozedur mit HyperSQL.

Organisation

Auch in der Organisation sollten Vorkehrungen getroffen werden, damit ETL-Entwicklungen mit PL/SQL erfolgreich sein können. Die vorgestellten Werkzeuge, die Bibliothek müssen gewartet und weiterentwickelt werden. Insbesondere vor dem Hintergrund der Zukunftsfähigkeit von Bibliothek, Metadaten und insbesondere den entwickelten ETLs ist es wichtig, klare Zuständigkeiten zu definieren. Für die folgenden Tätigkeiten ist es sinnvoll diese zu institutionalisieren:

- Entwicklungsrichtlinien und Templates
- Weiterentwicklung und Planung der Bibliothek
- Betrieb der Bibliothek
- Beschaffung bzw. Implementierung sowie Betrieb von Hilfsmitteln und Werkzeugen

Diese Rollen müssen nicht auf verschiedene Personen verteilt werden, von Bedeutung ist dabei dass die Zuständigkeit institutionalisiert wird. In der Organisation können hier verschiedenste Varianten gewählt werden. Im Folgenden werden drei Beispielszenarien knapp beschrieben.

Variante A: Eine kleine Gruppe in einem kleinem Umfeld kümmert sowohl um Entwicklung wie auch Betrieb von ETL und Bibliothek. In diesem Szenario sind alle Mitarbeiter für alle Tätigkeiten zuständig, diese Generalisten erledigen sämtliche anfallenden Aufgaben. Hier bestehen zwei Risiken: ein Risiko besteht darin, dass herkömmliche Lösungen sehr lange am Leben erhalten werden und nicht auf eine gemeinsame Metadatenbasis angepasst werden können. Das andere Extrem ist ein Overdesign einer Bibliothek, die letztlich auf Grund der Fülle der bereitstehenden Funktionalität nicht bedienbar ist und nicht gelebt werden kann.

Variante B: In einem großen Umfeld existiert eine eigene ETL Betriebsgruppe, die sich auch um die Bibliothek kümmert. Hier besteht bei großem Wartungsaufkommen die Gefahr, dass die Arbeit an der Bibliothek vernachlässigt wird. Hingegen kann Entwicklungspotential schnell erkannt werden, da täglich viele ETL betreut werden. Hier besteht das Risiko, dass Neuentwicklungen in der Bibliothek in erster Linie zu Gunsten des Betriebsteams entworfen werden, was nicht immer dazu führen muss dass diese auch im Entwicklungsbereich verwendet werden.

Variante C: Eine zentrale Architektenstelle kümmert sich um alle anfallenden Aufgaben rund um Bibliothek, Metadaten, Werkzeuge und Prozesse. Hier ist das Blickfeld oftmals weiter, es besteht Potential insbesondere darin, zusätzliche Metadaten in die Bibliothek zu integrieren. Es ist eine kontinuierliche Arbeit an der Bibliothek möglich. Dabei können auch Entwicklungen betrieben werden, die unabhängig davon sind ob evtl. zunächst entstehende Mehraufwände im Bereich

Entwicklung oder im Bereich Betrieb anfallen. Hier besteht das Risiko, dass Entwicklungen am Bedarf vorbeigehen oder nicht in Angriff genommen werden.

Zusammenfassung

Durch konsequente Berücksichtigung der Eigenschaften von ETLs mit PL/SQL wie Tool können bei der Verwendung von PL/SQL viele Vorteile erzielt werden. Durch Schaffung einer offenen Metadatenbasis sowie einer Bibliothek für gemeinsam genutzte Funktionen können Qualitätssteigerungen sowie Verbesserungen in Entwicklung und Betrieb erzielt werden. Die Ressourcen dazu sind leicht am Markt zu beschaffen. Dabei bleibt PL/SQL flexibel und offen für jede Form der Realisierung, minimiert die Abhängigkeiten bei Versionssprüngen der Datenbank und erlaubt für jeden Entwickler durchschaubares Tuning.
Der zentrale Punkt bei der Entwicklung von ETL Strecken mit PL/SQL ist die Schaffung einer gemeinsamen Metadatenbasis. Diese Metadaten müssen von zentral bereitgestellten Funktionen befüllt werden. Dabei ist es wichtig, ein einheitliches Datenmodell zu verfolgen. Die offene Gestaltung der Metadaten eröffnet die Möglichkeit, weitere Metadaten als nur aus ETL Prozessen zu integrieren. Damit die Nutzung, Weiterentwicklung und der Betrieb dieser Bibliothek stattfinden können, sind passende Ergänzungen an den entsprechenden Prozessen zur Softwareentwicklung erforderlich. Für die Weiterentwicklung der Bibliothek wird ein einfacher Prozess eingeführt. Die Zuständigkeiten für die Bibliothek, sowie für den Betrieb von Werkzeugen werden institutionalisiert. So kann sichergestellt werden, dass eine Weiterentwicklung nach tatsächlichen Bedürfnissen geschieht und ein Overdesign vermieden wird.
Damit ist es möglich, ETL-Strecken mit PL/SQL in hoher Qualität herzustellen und mit großer Stabilität zu betreiben. Die Komplexität der gewählten Lösungen ist dabei nur so hoch, wie sie im Rahmen der Bibliothek- und Metadatenstruktur notwendig sind. Weitere Lizenzkosten werden durch Verzicht auf ein ETL Tool vermieden.

Kontaktadresse:

Alexander Mendle
SHS VIVEON AG
Clarita-Bernhard-Str. 27
D-81249 München

Telefon: +49 162 2979176
Fax: +49 89 747257-900
E-Mail: Alexander.Mendle@SHS-VIVEON.com
Internet: www.shs-viveon.com