

# Konzeptionierung einer mehrschichtigen Haltung von Mess- und Qualitätsdaten

Ivonne Hutzler

Professor Dr.-Ing. Frank Herrmann

Ostbayerische Technische Hochschule Regensburg

E-Mail: frank.herrmann@oth-regensburg.de

## ABSTRACT

Bei der Zollner Elektronik AG fallen viele Daten bei der Abwicklung des Alltagsgeschäftes an. Die Nachverfolgbarkeit von Arbeitsabläufen wird immer wichtiger. Nicht nur gesetzliche Vorgaben erfordern die längerfristige Vorhaltung von Prozessdaten. Auch von Seiten der Kunden wird erwartet, dass Daten zur Kontrolle der Produktionsabläufe längerfristig gespeichert und zur Nachprüfung in aufbereiteter Form zur Verfügung gestellt werden. Daher soll die Fragestellung, wie eine mehrschichtige Haltung der Mess- und Qualitätsdaten bei der Firma Zollner Elektronik AG realisiert werden kann, behandelt werden. Hierzu wird zunächst im Rahmen einer Anforderungsanalyse geklärt, welche Leistungsmerkmale das Konzept zu erfüllen hat, um den Ansprüchen des Unternehmens, seiner Mitarbeiter und seiner Prozesse gerecht zu werden.

Anhand der ermittelten Anforderungen wird anschließend ein Konzept zur optimalen Umsetzung der Fragestellung aufgestellt. Hierzu werden mehrere Lösungsansätze vorgestellt und evaluiert. Abschließend wird eine Empfehlung zur möglichen Realisierung einer mehrschichtigen Haltung von Mess- und Qualitätsdaten bei Zollner Elektronik AG gegeben.

## Die Zollner Elektronik AG

Die Zollner Elektronik AG [Zoll16] gehört zu den weltweit führenden *EMS*-Dienstleistern (*Electronic Manufacturing Service*) und wurde 1965 vom heutigen Aufsichtsratsvorsitzenden Manfred Zollner sen. gegründet. Als Auftragsfertiger entwickelt und produziert das Unternehmen neben Einzelteilen und Modulen auch komplexe System- bzw. Gerätelösungen.

Mit dieser breiten Produktpalette beliefert die Zollner Elektronik AG unter anderem Unternehmen aus den Branchen Automotive, Luftfahrttechnik, Messtechnik oder Telekommunikation. Zum Kundenstamm gehören namhafte Unternehmen wie IBM, HP, Continental oder Bosch. Auch im Dienstleistungssektor ist das Unternehmen aktiv und bietet zusätzlich Supply Chain Management, Material-Management und After Sales Services an.

Neben dem Hauptsitz in Zandt (Oberpfalz, Bayern) und den Produktionsstandorten in Deutschland existieren weitere Standorte in Rumänien, China, Ungarn, Tunesien, Costa Rica, Hongkong, den USA und der Schweiz. Etwa die Hälfte der rund 10.000 Mitarbeiter ist in den acht bayerischen Niederlassungen beschäftigt. Der ehemalige Ein-Mann-Elektro- und Installationsbetrieb ist

heute der größte Arbeitgeber im Landkreis Cham. Die Zollner Elektronik AG befindet sich zu 100 Prozent in Familienbesitz.

## Ist-Zustand

Das Hauptwerk der Zollner Elektronik AG in Zandt und jedes Werk im Ausland führt seine eigene Oracle Datenbank zur Haltung der Daten, die während der Prozessabwicklung anfallen. Die Daten eines jeden Standortes werden in der sogenannten *Master-Datenbank*, die sich im Rechenzentrum im Hauptwerk in Zandt befindet, über Oracle Streams zusammengeführt. Abbildung 1 visualisiert diesen Aufbau. Die Struktur jeder Datenbank ist identisch und eine Replikation jeder Datenbank sorgt für Ausfallsicherheit und Wiederherstellbarkeit.

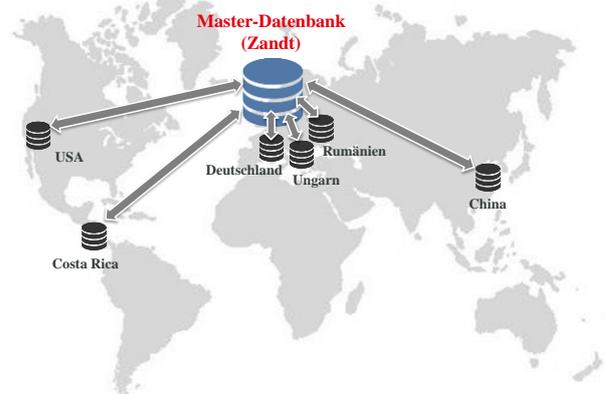


Abbildung 1: Werksübergreifende Datenbankstruktur

In dieser Arbeit wird lediglich der Ausschnitt der Datenbank-Struktur, der zur Verarbeitung von Mess- und Qualitätsdaten dient, betrachtet. Als Mess- und Qualitätsdaten werden die Daten verstanden, die von Testsystemen produziert werden. Testsysteme sind Maschinen, die zum Testen der hergestellten Produkte dienen. Zum Teil stellt das Unternehmen selbst entsprechende Testsysteme her, jedoch kommen auch zugekaufte Testsysteme von anderen Herstellern zum Einsatz.

Es gibt mehrere Arten von Prüfverfahren. Grob lassen diese sich in die Kategorien *AOI/X-Ray*, *Funktionstests*, *In-Circuit-Tests*, *Sicherheitstests* und *Sondertests* unterteilen. Zu den am häufigsten durchgeführten Prüfverfahren bei Produkten der Zollner Elektronik AG gehören unter anderem der *FAT* (*Function Analyse Test*), der *ICT* (*In-Circuit-Test*), sowie die *AOI* (*Automatic Optical Inspection*). Beim *FAT* wird die Funktionalität einer

Baugruppe getestet. Funktionalität bezieht sich in diesem Kontext auf das Bedienen von Schaltern und den korrekten Ablauf programmierter Funktionen eines Gerätes. Der *In-Circuit-Test* ist ein Prüfverfahren für elektronische Baugruppen bzw. Schaltkreise (engl. *electrical circuit*) und bestückte Leiterplatten in der Elektronikfertigung. Beim *ICT* steht die Prüfung der Bauelementparameter einer bestückten Baugruppe oder der elektrischen Verbindungen einer Leiterplatte im Vordergrund. Dabei wird die bestückte Leiterplatte, nachdem sie auf bzw. in einen speziellen Prüfadapter gelegt wurde, auf Fehler in der Leiterbahnführung, Lötfehler und Bauteilefehler geprüft. Auch ganze Schaltungsblöcke (Cluster) können getestet werden. Die *AOI* entspricht einer Sichtprüfung, die entweder von Personen oder Maschinen während der Produktion elektronischer Baugruppen durchgeführt wird, um die fehlerfreie Herstellung sicherzustellen. Beim Bestücken durch Bestückungsautomaten oder beim anschließenden Lötprozess können Fehler auftreten. Fehler können beispielsweise fehlende oder falsch bestückte Bauteile, verdrehte oder versetzte Bauteile oder Verunreinigungen der Leiterplatte sein. Diese Fehler müssen vor dem nächsten Arbeitsschritt in der Fertigung abgefangen und – sofern möglich – behoben werden.

Die Testsysteme sind in den Produktionsprozess integriert. Oft muss während eines Testablaufes auch ein Produktionsschritt oder gegebenenfalls eine Reparatur durchgeführt werden. Typische Testarbeitsplätze für elektronische Bauteile bei der Zollner Elektronik AG bestehen aus einem Testgerät, einem angeschlossenen Monitor und dem benötigten Werkzeug für die durchzuführenden Test- und Arbeitsschritte. Zum Teil ist auch Werkzeug für eventuelle Reparaturen vorhanden. Ein Mitarbeiter der zuständigen Abteilung – das Prüfpersonal bzw. ein Produktionsmitarbeiter – bedient das Testsystem. Nachdem das Bauteil in den Tester gelegt wurde, wird der Testvorgang mit Hilfe des *MCP* (*Master Control Program*) gestartet. Als erstes muss ein Barcode gescannt werden, der das Bauteil eindeutig identifiziert. Mit Hilfe dieser Information wird das durchzuführende Testprogramm vom Server geladen. Die Testprogramme wurden zuvor mittels *LabView*<sup>1</sup> (*Laboratory Virtual Instrumentation Engineering Workbench*) oder entsprechender Vorgängerprogramme von Mitarbeitern der zuständigen Abteilung erstellt. Wurde die Testsoftware auf dem Server gefunden, wird diese gestartet. Ab hier läuft das Testprogramm automatisch ab. Während eines jeden Teilschrittes, wird dem Prüfpersonal visualisiert, was derzeit getestet wird und je nach Programm auch die noch verbleibende Dauer. Auf den Bestückungsanlagen befinden sich digitale Bauanleitungen zur Hilfestellung während der Montage. Zum Teil werden die Bauteile hierbei auch programmiert.

Testsysteme geben ihre Daten in Form von Testprotokollen aus. Diese Testprotokolle liegen in unterschiedlichen Formaten vor. Bevor die Testprotokolle verarbeitet werden können, müssen diese gegebenen-

falls in eines der beiden Standard-Formate der Zollner Elektronik AG konvertiert werden. Das Format *zvei* stellt eine *XML*-Struktur dar, *pdm* eine strukturierte *ASCII*-Zeichenfolge. Ersteres ist eine Anlehnung an das ehemals verwendete Format einer früheren Software zur Prüfdatenverwaltung. Zweiteres ist eine Festlegung der Zollner Elektronik AG zur Darstellung der Daten, die dadurch entstanden ist, dass viele Testsysteme keine standardisierte Ausgabe besitzen. Beispieldateien für beide Formate befinden sich im Anhang unter **Fehler! Verweisquelle konnte nicht gefunden werden.**

Nach der Generierung und gegebenenfalls Konvertierung der Protokolle, werden die Testprotokolle auf einem Netzlaufwerk abgelegt. Dies geschieht zum Teil per Software, zum Teil aber auch manuell durch das Prüfpersonal. Dadurch ist nicht gewährleistet, dass bereits erstellte Protokolle auch immer sofort auf dem Server zu finden sind. Es kann vorkommen, dass Protokolle zwar lokal auf dem Testsystem erstellt, jedoch nie abgeschickt werden. Nach Ablage der Testprotokolle werden diese mit Hilfe eines unternehmensintern programmierten Parsers analysiert. Bei Fehlern wird das Dokument als fehlerhaft markiert und zurückgewiesen. Die Fehler müssen dann manuell behoben werden. Wurden die Fehler behoben, wird das Dokument erneut mit dem Parser analysiert. Fehlerfreie Protokolle werden in die Datenbank importiert und die enthaltenen Testdaten entsprechend dem Qualitäts-Daten-Schema gespeichert. Abbildung 2 zeigt den Prozess der Übermittlung und Speicherung der Testdaten.

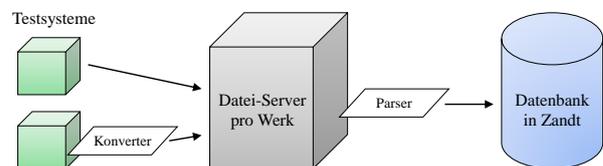


Abbildung 2: Ablageprozess der Testdaten

Die Erfassung und Verarbeitung von Testdaten ist bereits seit Firmengründung ein fester Bestandteil der Qualitätssicherung bei der Zollner Elektronik AG. Hierzu wurden über die Jahre unterschiedliche Systeme verwendet. In den Anfängen arbeitete die Firma mit einem zugekauften System. Dieses wurde durch die Eigenentwicklung des *PDE* (Prüfdatenerfassung) abgelöst und zum *PDM* (Prüfdatenmanagement) weiterentwickelt. Aufgrund der wachsenden Anforderungen, sowohl von Seiten der Kunden, als auch durch rechtliche Vorgaben, wurde im Jahre 2006 das Softwareprodukt *intraFACTORY*® der Kratzer Automation AG als zentrale *MES*-Plattform (*Manufacturing Execution System*) eingeführt. Kratzer hat sich jedoch 2011 aufgrund mangelnder Profitabilität dazu entschlossen, *intraFACTORY*® nicht mehr als Serienprodukt, sondern nur noch als „individuelle *MES*-Lösung für die Elektronikindustrie“ anzubieten [Zühl11]. Aus diesem Grund haben die Verantwortlichen der Zollner Elektronik AG beschlossen, die vorhandene *MES*-Lösung in Zukunft selbst weiter zu entwickeln. Seitdem wird schrittweise die

<sup>1</sup> grafisches Programmiersystem

genutzte Funktionalität aus dem *intraFACTORY*® durch selbstprogrammierte Anwendungen abgelöst. Einige Module wurden bereits in das neu entwickelte Zollner-MES-Framework integriert. Noch nicht abgelöste Funktionalitäten von *intraFACTORY*® befinden sich bis zur kompletten Ablösung des Legacy Systems weiterhin in Gebrauch. Die Datenbankstruktur, auf die das Zollner-MES-Framework aufbaut, umfasst derzeit 265 Relationen und weist eine Größe von etwa zwei Terabyte auf. Über die Relation *ORDERS* (Auftrag), sind sämtliche Tabellen miteinander verbunden.

### Qualitäts-Daten-Schema (Q-Daten-Schema)

Zur Speicherung von Mess- und Qualitätsdaten erarbeitete die Zollner Elektronik AG auf Basis der bestehenden Datenstruktur des zugekauften Systems ein Qualitäts-Daten-Schema (Q-Daten-Schema). Dieses Schema wurde in die bestehende Datenbasis integriert.

Abbildung 3 zeigt den Ausschnitt der Relationen aus diesem System, sowie deren Beziehungen untereinander. Die mit *P* markierten Spalten stellen dabei den Primärschlüssel der jeweiligen Tabelle dar. Das *F* vor einer Spalte signalisiert, dass es sich dabei um einen Fremdschlüssel handelt. Die Spalten, die mit einem roten Stern markiert sind, dürfen nicht *null* sein.

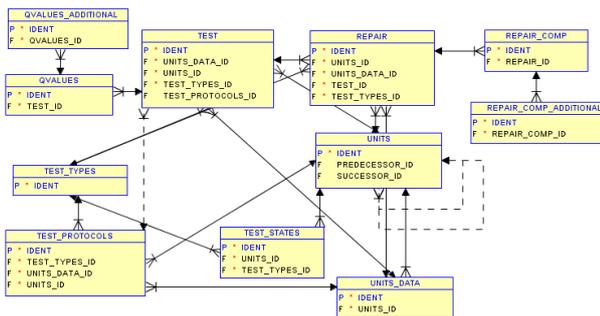


Abbildung 3: Überblick über die Relationen und Beziehungen des Q-Daten-Schemas

Die Relation *UNITS* beinhaltet alle Elemente (Baugruppen), die von der Zollner Elektronik AG hergestellt werden. Die Tabelle *UNITS\_DATA* enthält einen Eintrag für jede Station, die eine bestimmte Unit bereits durchlaufen hat. In der Tabelle *TEST\_PROTOCOLS* werden die komprimierten Testprotokolle mit Erstzeit und Status abgelegt. Die aus den Protokollen ermittelten Testdaten werden in der Tabelle *TEST* gespeichert. Jeder Test besitzt eine Start- und Endzeit, ein Testergebnis und eine zugehörige Test-Software. Tests bestehen oft aus mehreren Teilschritten (Subtests). Die Subtests und die zugehörigen Detail-Informationen finden sich in der Relation *QVALUES* wieder. Die Relation *QVALUES\_ADDITIONAL* ist zur Speicherung zusätzlicher optionaler Informationen zu den einzelnen Testschritten vorgesehen. Die Tabelle *TEST\_TYPES* enthält mehrere Arten von Tests zur Kategorisierung der Prüfverfahren. In *TEST\_STATES* wird der jeweils aktuelle Status des Baugruppentests abgelegt. Reparaturen enthalten eine Start- und Endzeit und können in Teilrepara-

turen unterteilt werden. Reparaturdaten zu einzelnen Bauteilen werden in der Tabelle *REPAIR\_COMP* gespeichert. Ebenso wie bei der Relation *QVALUES* gibt es auch hier eine Tabelle zur Speicherung zusätzlicher optionaler Informationen, die Tabelle *REPAIR\_COMP\_ADDITIONAL*.

### PDM-Datenmodell

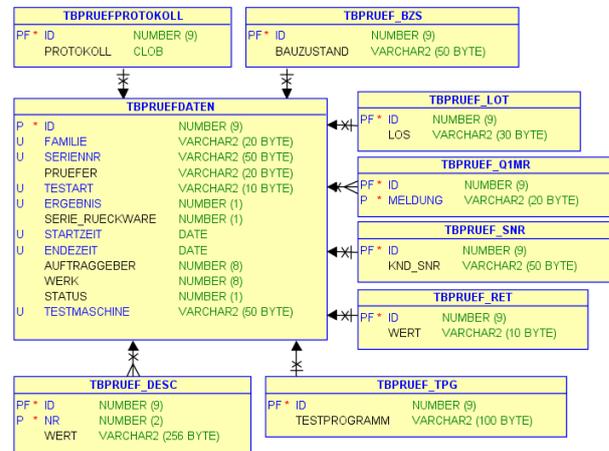


Abbildung 4: PDM-Datenmodell

Das Datenmodell des PDM ist im Gegensatz zum Zollner-MES sternförmig aufgebaut. Abbildung 4 visualisiert die wichtigsten Tabellen aus diesem Schema. Die Tabelle *TBPRUEFDATEN* persistiert alle Pflichtdaten, die im Zusammenhang mit Tests abgelegt werden müssen. Für die optionalen Attribute existiert jeweils eine eigene Relation, die jeweils einen Fremdschlüssel auf die Tabelle *TBPRUEFDATEN* besitzt. Die Relation *TBPRUEF\_BZS* dient der Speicherung des Bauzustandes eines Elements. Die Tabelle *TBPRUEF\_SNR* wird zur Speicherung der Kundennummer verwendet. Prüfprotokolle werden in der Tabelle *TBPRUEFPROTOKOLL* persistiert. Die Relation *TBPRUEF\_RET* speichert gegebenenfalls den Typ der Rückware. Die Relation *TBPRUEF\_DESC* dient zur Speicherung von bis zu zehn zusätzlichen Beschreibungen. Zur Speicherung von Losen<sup>2</sup> wird die Tabelle *TBPRUEF\_LOT* verwendet. Die Tabelle *TBPRUEF\_QIMR*<sup>3</sup> dient der Persistierung von Meldungen (Nachrichten). Zur Speicherung der zu dem/den Test/s gehörigen Testprogramm/en dient die Relation *TBPRUEF\_TPG*.

Die mit *U* gekennzeichneten Felder in der Relation *TBPRUEFDATEN* bilden zusammen einen eindeutigen (unique) Schlüssel. Die mit *P* gekennzeichneten Spalten stellen jeweils die Primärschlüssel der Spalte dar. Die mit *PF* gekennzeichneten Spalten zeigen an, dass es sich dabei sowohl um einen Primär- als auch einen Fremdschlüssel handelt. Bei Tabellen, bei denen mehrere Spalten mit *P* und/oder *PF* gekennzeichnet sind, bilden diese jeweils zusammen den Primärschlüssel.

<sup>2</sup> frühere Bezeichnung für einen Fertigungsauftrag

<sup>3</sup> Bezeichnung angelehnt an SAP-Nummerierung für Meldungen

Zur schnelleren Auswertung der Daten wurden zwei weitere Tabellen mit verdichteten Daten hinzugefügt. Diese sind in Abbildung 5 dargestellt. Unter Verdichtung wird in diesem Kontext das Zusammenfassen von Daten nach bestimmten Kriterien verstanden. Beispielsweise können diese Kriterien Zeit oder eine bestimmte Serien- oder Kundennummer darstellen. In der Tabelle *TBMWA* ist die Anzahl der fehlerhaft durchgeführten Tests pro Baugruppe pro Tag abgelegt. Die Tabelle *TBMWA\_SERIE* speichert die Gesamtanzahl aller erfolglos, sowie erfolgreich durchgeführten Test, die pro Testschritt, Produktfamilie und Testmethode pro Tag durchgeführt wurden. Jedoch werden hier nur die Tests berücksichtigt, die zum ersten Mal an dieser Baugruppe ausgeführt wurden. Die Inhalte der Tabellen wurden hier so gewählt, dass damit ein paar der häufigsten Abfragen auf die Daten in diesem Schema performanter abgearbeitet werden können.

TBMWA		TBMWA_SERIE	
P * FAMILIE	VARCHAR2 (50 BYTE)	P * FAMILIE	VARCHAR2 (50 BYTE)
P * BZS	VARCHAR2 (50 BYTE)	P * BZS	VARCHAR2 (50 BYTE)
P * TEST	VARCHAR2 (255 BYTE)	P * TEST	VARCHAR2 (255 BYTE)
P * TAR	VARCHAR2 (20 BYTE)	P * TAR	VARCHAR2 (20 BYTE)
P * TMA	VARCHAR2 (50 BYTE)	P * TMA	VARCHAR2 (50 BYTE)
P * EDATUM	DATE	P * EDATUM	DATE
COUNTER	NUMBER	CP	NUMBER (9)
		CF	NUMBER (9)

Abbildung 5: PDM-Tabellen für Verdichtung

Mit Hilfe der im PDM enthaltenen Abfrage- und Filtermöglichkeiten, können nahezu alle Abfragen getätigt werden, die auch direkt auf dem Datenbanksystem zur Auswertung der Test- und Reparaturdaten möglich wären. Diese Abfragemöglichkeiten, sind für das neue Datenschema noch nicht umgesetzt. Auch enthält das Zollner-MES bislang keine verdichteten Tabellen.

### Problemstellung

Mit dem stetigen Wachstum der Mess- und Qualitätsdaten bei der Zollner Elektronik AG verlangsamt sich zunehmend sowohl das Einfügen neuer Datensätze, als auch das Auslesen der Daten aus dem MES-System. Die abgelegten Testdaten werden jedoch nur begrenzt lange benötigt. Sobald ein Produkt fertig getestet und vertrieben ist, werden die abgespeicherten Daten lediglich noch in Ausnahmefällen benötigt, beispielsweise wenn ein Kunde in der Gewährleistungszeit einen Fehler am gelieferten Produkt reklamiert. Aufgabe ist nun zu ermitteln, wie hier eine sinnvolle Trennung der Daten erfolgen könnte.

### Mehrschichtige Datenhaltung

Die beiden zu lösenden Hauptprobleme des Systems sind die Performancesteigerung bei Abfragen und die Auslagerung der nicht mehr benötigten Daten aus dem Produktivsystem. Um diese Probleme bestmöglich beheben zu können, wird der Konzeptentwurf in mehrere Schichten unterteilt.

Mehrschichtigkeit bedeutet in diesem Fall nicht etwa eine Aufteilung der Datenbank in mehrere gleich strukturierte Datenbanken, die alle jeweils nur Daten über

einen bestimmten Zeitraum beinhalten (Drei-, Sechs-, Zwölf-Monats-Datenbank), sondern die Aufteilung der Daten in drei logische Schichten/Ebenen. Dabei beinhaltet die erste Ebene die Produktiv-Daten. Diese befinden sich in der Produktiv-Datenbank und stellen die bisherige Datenbasis dar. Diese Daten sollen mit Hilfe der weiteren Ebenen unterteilt werden. Die zweite Ebene stellt eine Datenverdichtung dar. Hierbei sollen die Daten so aufbereitet werden, dass sie effizient abgefragt werden können. Die dritte Ebene dient der Datenarchivierung. In das Archiv sollen alle Daten ausgelagert werden, die nicht mehr im Produktivbetrieb benötigt werden. Abbildung 6 visualisiert diese Unterteilung.

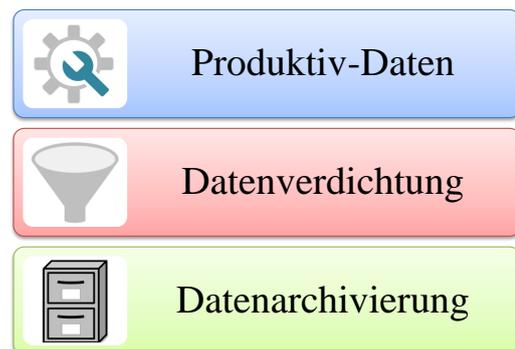


Abbildung 6: Ebenen der mehrschichtigen Datenhaltung

Aufgrund der mehrschichtigen Datenhaltung, lassen sich auch die Anforderungen den beiden zusätzlichen Schichten zuordnen.

### Anforderungen bei der Datenverdichtung

- Bei statistischen Auswertungen (insbesondere Auswertungen über mehrere Kalenderwochen oder Monate) soll eine Performancesteigerung erreicht werden.
- Der Verwaltungsaufwand soll gering gehalten werden.
- Die Neuanschaffungskosten, sowie die laufenden Kosten sollen gering sein.
- Bestehende Systeme und Tools sollen ohne großen Änderungsaufwand weiter verwendet werden können.
- Alle Abfragen, die im PDM bereits existieren, sollen später auch im MES zur Verfügung stehen.
- Das Konzept soll weitestgehend herstellerunabhängig sein.
- Die Zugriffszeiten auf die Produktiv-Datenbank sollen verringert werden.

### Anforderungen bei der Datenarchivierung

- Sollten Daten aus dem Produktivsystem extrahiert und in ein externes System ausgelagert werden müssen, sollen diese bei Bedarf auch wieder zurückgeführt werden können.
- Der Verwaltungsaufwand soll gering gehalten werden.

- Die Neuanschaffungskosten, sowie die laufenden Kosten sollen gering sein.
- Der Speicherplatzbedarf im Produktivsystem soll verringert werden.
- Die Zugriffszeiten auf die Produktiv-Datenbank sollen verringert werden.

### Konzept-Entwurf

Wie bereits zuvor erläutert, unterteilt sich das Konzept in mehrere Ebenen. Im Folgenden werden für die Ebenen *Datenverdichtung* und *Datenarchivierung* jeweils mehrere Lösungsansätze untersucht und diese anhand der gestellten Anforderungen evaluiert.

### Datenverdichtung

Die Datenverdichtung stellt die zweite Ebene der Datenhaltung dar. Sie ist hauptsächlich auf schnelle Abfragen und Auswertungen hin zu konzipieren. Darüber hinaus sollen die weiteren zuvor aufgelisteten Anforderungen berücksichtigt werden. Zur Bewältigung dieser Anforderungen, werden als Lösungsansätze die Partitionierung von Datenbanktabellen, die Verwendung von Materialisierten Sichten und die Nutzung eines Data-Warehouses vorgestellt. Eine Performancesteigerung durch Verbesserung der Hardware ist auszuschließen, da bei der Zollner Elektronik AG bereits sehr leistungsfähige Hardware im Einsatz ist. Eine Betrachtung der Indizes wird ebenfalls nicht vorgenommen. Die Indizes belaufen sich derzeit auf etwa drei Terabyte an Daten. Dies übersteigt deutlich die Menge der Nutzdaten. Aus diesem Grund wird die Betrachtung von Indizes separat vorgenommen und in dieser Arbeit nicht weiter betrachtet.

### Partitionierung

In diesem Abschnitt wird die Verwendung einer Partitionierung zur Erfüllung der zuvor dargelegten Anforderungen an die Datenverdichtung untersucht.

### Performancesteigerung bei statistischen Auswertungen

Partitionierung kann zur Verbesserung der Performance beitragen. Durch Partitionierung der großen Tabellen des Schemas (*UNITS\_DATA*, *QVALUES*) könnte ein schnellerer Zugriff auf die Daten erfolgen, sofern diese entsprechend der gewählten Partitionierung abgefragt werden. Hierbei könnte die Performance sowohl im transaktionalen Betrieb der Datenbank, als auch bei komplexen Abfragen gesteigert werden. Die Performancesteigerung durch Partitionierung ist dabei jedoch sehr stark von dem zugrundeliegenden Partitionierungskonzept abhängig. Werden Daten nicht entsprechend der gewählten Partitionierung abgefragt, kann dabei auch keine Performancesteigerung erreicht werden. Um eine Abschätzung der Performancesteigerung geben zu können, wird nachfolgend ein Beispiel genauer erläutert.

Angenommen die Tabelle *QVALUES* wäre mittels *Range-List*-Partitionierung umgesetzt. Tageweise könnte diese nach der *Range*-Partitionierung mit Partitionierungsschlüssel *ENDTIME* partitioniert sein. Zudem könnten Sub-Partitionen nach der *List*-Partitionierung mit *TEST\_ID* als Partitionierungsschlüssel angelegt sein.

Hierdurch könnten Abfragen beschleunigt werden, die eine tageweise Einschränkung oder/und eine Beschränkung auf einen bestimmten Test aufweisen, da nicht mehr die gesamten Daten abgefragt werden müssten, sondern lediglich die Partitionen, die die entsprechenden Daten beinhalten.

Durch eine derartige Partitionierung wären deutlich weniger Datensätze zu lesen. In dem konkreten Beispiel würden jedoch ohnehin die bereits existierenden Indizes herangezogen, wodurch lediglich eine geringe Performancesteigerung zu erwarten wäre.

### Verwaltungsaufwand

Partitionierung erfordert etwas Verwaltungsaufwand. Wie bereits erklärt, muss zunächst ein Partitionierungskonzept entwickelt werden, das auf das Zugriffsverhalten angepasst ist. Die Wahl der passenden Partitionierungsstrategie hat einen hohen Einfluss auf die Größe der Leistungsveränderungen [Brot03]. Für die Ersteinrichtung und Konzepterstellung der Partitionierung in diesem konkreten Fall, wird ein Zeitaufwand von etwa zehn Arbeitstagen angenommen. Der laufende Administrationsaufwand wird auf etwa einen Arbeitstag pro Monat geschätzt. Bei den genannten Werten handelt es sich um Schätzungen der zuständigen Mitarbeiter der Abteilung zur Administration der Datenbankstruktur bei der Zollner Elektronik AG.

### Kosten

Die Zollner Elektronik AG hat derzeit zehn Oracle Datenbanken in der Enterprise Edition inklusive des *Diagnostic*- und *Tuning-Packs* im Einsatz. Die Partitionierung stellt eine zusätzliche Option dar, die zu den genannten Optionen hinzugebucht werden müsste. Hierbei würden zusätzliche Lizenzkosten gemäß dem Oracle-Lizenzmodell anfallen [Orac16]. Die Lizenzkosten berechnen sich nach der Anzahl der verwendeten Prozessoren. Für jeden Prozessor, für den die gewünschte Option freigeschaltet werden müsste, würde ein bestimmter Betrag zur Freischaltung anfallen. Darüber hinaus fallen jährliche Kosten für Software-Update-Lizenzen und den Support an. Auf Basis dieser Berechnungsgrundlage würden sich die Erstanschaffungskosten auf etwa

$$10 \text{ Prozessoren} * 11.500 \text{ US Dollar} \\ = \mathbf{115.000 \text{ US Dollar}}$$

belaufen. Für die Software-Update-Lizenzen und den Support würden zudem jährlich Kosten in Höhe von etwa

$$10 \text{ Prozessoren} * 2.530 \text{ US Dollar} \\ = \mathbf{25.300 \text{ US Dollar}}$$

anfallen.

## Wiederverwendbarkeit bestehender Systeme

Die bestehenden Systeme, wie beispielsweise Tools zur Abfrage der Daten aus dem MES, könnten ohne Anpassung weiter verwendet werden. Dies liegt vor allem daran, dass sich die Datenbank selbst um die Verwendung der richtigen Partitionen der jeweiligen Tabelle kümmert. *SQL*-Anfragen könnten somit wie gewohnt gestellt werden, da die Abfragen automatisch durch das Datenbankmanagementsystem (DBMS) angepasst werden.

## Erweiterung des MES um Abfragemöglichkeiten des PDM

Wie auch bei den bestehenden Systemen, gibt es bei der Erweiterung des MES um weitere Abfragemöglichkeiten durch die Verwendung der Partitionierung keinerlei Einschränkungen.

## Herstellerunabhängigkeit

Da als Datenbank die *Oracle DB* verwendet wird, ist es notwendig, auch die Option der Partitionierung von Oracle zu verwenden. Sollte jedoch der Datenbank-Hersteller gewechselt werden, gibt es auch andere Anbieter, die Partitionierung anbieten. Hierbei gibt es jedoch Unterschiede bei der Verfügbarkeit der verschiedenen Varianten der Partitionierung. Die folgende Tabelle zeigt die möglichen Partitionierungsmöglichkeiten in Oracle, SQL Server und MySQL. [Schi14]

Hersteller	Verfahren zur Partitionierung
Oracle	Bereichs-Partitionierung List-Partitionierung Hash-Partitionierung Intervall-Partitionierung Referenz-Partitionierung virtuelle spaltenbasierte Partitionierung
SQL Server	Bereichs- Partitionierung Index- Partitionierung
MySQL	Bereichs-Partitionierung List-Partitionierung Hash-Partitionierung Schlüssel-Partitionierung

Tabelle 1: Unterstützung der Partitionierung in Datenbanken nach [Schi14]

## Zugriffszeiten auf die Produktiv-Datenbank

Wie bereits im Abschnitt *Performancesteigerung bei statistischen Auswertungen* erläutert, können nicht nur komplexe Abfragen auf die Datenbank, sondern auch die transaktionalen Anfragen an die Datenbank beschleunigt werden. Zudem wird bei Manipulationen der Tabelle kein exklusiver Zugriff auf die gesamte Tabelle benötigt, sondern lediglich für die Partition(en), in der der Datensatz/die Datensätze manipuliert werden soll(en). Damit können unter Umständen Zugriffe parallelisiert werden.

## Zusatztabellen im Produktivsystem mit aggregierten Daten

Im PDM wurde bereits eine Verdichtung mittels zwei zusätzlicher Tabellen mit aggregierten Daten umgesetzt. Deshalb ist zu prüfen, ob diese Variante der Verdichtung auf das MES übertragbar ist. Hierzu würden in der Produktiv-Datenbank zusätzliche Tabellen mit verdichteten Daten nach dem Leitbild des PDM angelegt und verwaltet werden. Die Durchführung der Verdichtung im PDM ist bislang jedoch aufgrund der großen zu verarbeitenden Datenmengen sehr zeitintensiv. Über einen Zeitraum von ca. zwei Wochen ergibt sich eine durchschnittliche tägliche Berechnungsdauer von etwa 205 Minuten. Als Vergleichsbasis zeigt Abbildung 7 die zu der Zeit angefallenen Prüfprotokolle. Hierbei ist zu erkennen, dass an Wochenenden deutlich weniger Prüfprotokolle anfallen.

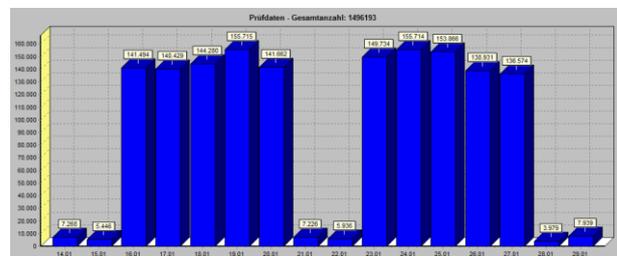


Abbildung 7: Anzahl importierter Prüfprotokolle

Zudem beinhaltet das PDM weniger Datensätze als das MES. Eine derartige Verdichtung wäre im MES somit nur umsetzbar, wenn eine Performancesteigerung bei der Durchführung der Verdichtung erreicht werden könnte.

Um zu ermitteln, welche Daten wie verdichtet werden sollen, werden zunächst die Abfragen analysiert, die derzeit im PDM zur Ermittlung der verdichteten Daten verwendet werden. In diesen Tabellen wird festgehalten, welche Testdurchläufe an einem bestimmten Tag, in einer speziellen Kalenderwoche oder in einem bestimmten Monat bei einem Kunden an dessen zu produzierenden Elementen erfolgreich und welche nicht erfolgreich durchgeführt wurden. Dabei wird unterschieden, ob der Test an diesem Element bereits auf diesem Testsystem durchgeführt wurde, oder ob er zum ersten Mal an diesem Element durchgeführt wird. Der Zeitraum, über den verdichtet werden soll, beschränkt sich auf einen Tag. Eine tageweise Verdichtung kann anschließend wieder gruppiert zu Kalenderwochen und Monaten für Auswertungen über längere Zeiträume herangezogen werden. Eine weitere Haltung zusätzlicher Tabellen für die nach Kalenderwochen und Monaten gruppierten Daten, wäre damit nicht erforderlich. Bei der Verdichtung sind die konkreten Messwerte der Tests nicht von Bedeutung, sondern lediglich der Erfolg bzw. Misserfolg eines Tests. Um eine äquivalente Abfrage im MES durchführen zu können, werden die Tabellen *QVALUES*, *TEST*, *TEST\_TYPES* und *PRODUCTS* benötigt.

Zur Verknüpfung der Daten der Tabelle *PRODUCTS*, werden zudem die Relationen *ORDERS* und *UNITS\_DATA* benötigt. Abbildung 8 visualisiert die Abhängigkeiten der Relationen.

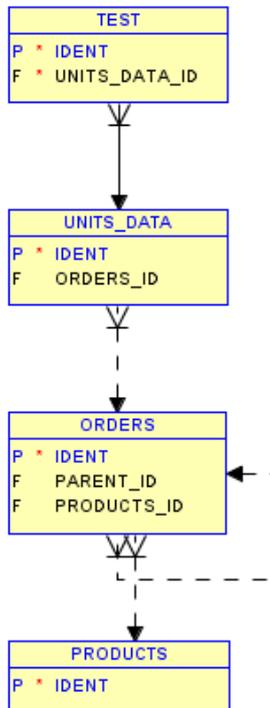


Abbildung 8: Verknüpfung der Relation *PRODUCTS* mit *TEST*

Die Darstellung entspricht der zuvor erläuterten Notation. Für die Tabelle der verdichteten Daten würden Spalten für den Testtag, die Artikelnummer, den Test-Typ, den genauen Zeitstempel des spätesten zusammengefassten Tests, die Anzahl der erfolgreichen Tests, sowie die Anzahl der fehlerhaften Tests angelegt werden. Die folgende Abbildung zeigt, wie die Abfrage aussehen könnte.

```

1  select DISTINCT
2  TO_CHAR(t.endtime, 'YY-MM-DD') Testday,
3  p.barcode as Product,
4  tt.barcode as Test_Type,
5  q.sub_test_name,
6  MAX(t.endtime) as EndTime,
7  Count(*) as Count,
8  sum(decode (q.SUB_TEST_RESULT, 1,1,0))
9  ok,
10 sum(decode (q.SUB_TEST_RESULT, 0,1,0))
11 failed
12 from qvalues q
13 inner join test t on t.ident = q.test_id
14 inner join test_types tt
15   on tt.ident = t.test_types_id
16 inner join units_data ud
17   on t.units_data_id = ud.ident
18 inner join orders o
19   on o.ident = orders_id
20 inner join products p
21   on p.ident = o.products_id
22 group by TO_CHAR(t.endtime, 'YY-MM-DD'),
23 p.barcode, tt.barcode, q.sub_test_name;
  
```

Abbildung 9: Beispielhafte Abfrage der Daten zur Verdichtung

Mit diesem Befehl könnten alle benötigten Daten abgefragt werden, die in der Verdichtungs-Tabelle enthalten sein sollen. Diese Tabelle könnte technisch betrachtet als Materialisierte Sicht realisiert oder mittels *SQL*-Prozedur oder Ähnlichem manuell erstellt und verwaltet werden. Eine einfache Möglichkeit zur Generierung der verdichteten Tabellen, stellt die Verwendung einer Materialisierten Sicht dar. Hierbei wäre der Aktualisierungs-Modus *FAST* zu verwenden. Die verdichteten Daten würden dabei nicht in eine separate Datenbank ausgelagert, sondern in der Produktiv-Datenbank gehalten.

Bei der Ausführung dieser Abfrage wären neben den Relationen auch die entsprechenden Indizes zu lesen. Dieser Prozess würde sehr viel Zeit und Speicher benötigen, da hier zur Fehlerfreiheit alles in einer geschlossenen Transaktion erfolgen müsste. Die Größe dieser Basistabellen und Indizes belaufen sich zusammen auf etwa 1953 Gigabyte (Stand: 16.01.2017). Durch Ausführung dieser Sicht würden ohne Einschränkung etwa zwei Terabyte an Daten gelesen werden, wodurch voraussichtlich der *TEMP-Tablespace* überlaufen würde, da dieser lediglich 88 Gigabyte umfasst.

Zusätzlich zum Schreiben der Materialisierten Sicht werden *Redo-Logs* geschrieben. Die Datenmenge der *Redo-Logs* entspricht etwa der Datenmenge der Materialisierten Sicht und den Indizes, die wiederum zu der Materialisierten Sicht angelegt würden.

Allein das einmalige Anlegen der Daten wäre somit bereits eine große Belastung für das Produktivsystem. Zudem gibt es Einschränkungen beim Refresh-Modus *Fast*, was im schlimmsten Fall dazu führen kann, dass die Materialisierte Sicht erneut komplett berechnet werden müsste. Durch mehrmalige Ausführung der Sicht erhöht sich auch die Datenmenge der *Redo-Logs*, was auch hier zu einem Überlauf führen kann.

Auch wenn es ausreichend ist, die Sicht nur im Master anzulegen und zu verwalten, da Abfragen immer auf die Masterdaten erfolgen, würde das Produktivsystem extremer Belastung ausgesetzt. Wie das obige Beispiel zeigt, ist eine Verdichtung in Form einer Materialisierten Sicht nur dann sinnvoll, wenn über wenige Tabellen aggregiert wird.

### Performancesteigerung bei statistischen Auswertungen

Durch die Verdichtung mittels Materialisierter Sicht, könnte ein enormer Performancegewinn für die jeweiligen Abfragen erreicht werden. Wie bereits im PDM gezeigt, dauert die Berechnung der gewünschten Abfrage im Schnitt pro Tag etwa 3,4 Stunden. Würde keine Materialisierte Sicht hierfür angelegt werden, würde jede Abfrage diese Zeit beanspruchen. Die benötigte Zeit bei Abfragen kann durch Verwendung der Vorberechnung auf die Zeitdauer für das Auslesen der Tabelle beschränkt werden.

## **Verwaltungsaufwand**

Diese Art der Verdichtung fordert sehr viel Verwaltungsaufwand. Zunächst müssen alle Abfragen ermittelt werden, die als Materialisierte Sicht umgesetzt werden müssten, um die wichtigsten Abfragen vorzuhalten. Es müsste eine Routine zur Ermittlung der jeweiligen Daten und ein entsprechender Update-Mechanismus erstellt werden. Hinzu kommt, dass diese Variante sehr pflegeintensiv ist und regelmäßig, besonders in der Anfangsphase auf mögliche Fehler geprüft werden muss.

## **Kosten**

Durch die Verwendung zusätzlicher Tabellen zur Aggregation von Daten würden keinerlei zusätzliche Kosten entstehen. Dies lässt sich dadurch begründen, dass weder neue Hardware angeschafft werden müsste, noch zusätzliche Lizenzkosten anfallen würden.

## **Wiederverwendbarkeit bestehender Systeme**

Die bisherigen Systeme könnten ohne größere Anpassung weiter verwendet werden. Hierzu müsste lediglich eine Lösung innerhalb der Datenbank geschaffen werden, durch die bei Anfragen an das System, automatisch die verdichteten Tabellen verwendet werden würden.

## **Erweiterung des MES um Abfragemöglichkeiten des PDM**

Wie bei der Partitionierung, gibt es auch hier keinerlei Einschränkungen bei der Erweiterung bestehender Systeme um zusätzliche Abfragen. Jedoch ist bei zusätzlich abzudeckender Funktionalität abzuwägen, ob es sinnvoll ist, zusätzliche aggregierte Daten in weiteren Tabellen zu halten.

## **Herstellerunabhängigkeit**

Das Prinzip der Verdichtung mittels zusätzlicher Tabellen zur Persistierung aggregierter Daten ist herstellerunabhängig, da diese Funktionalität, beispielsweise auch unter MSSql als indizierte Sicht, zur Verfügung steht. Hierbei gibt es jedoch Unterschiede bei der konkreten Umsetzung und den Automatisierungsmöglichkeiten. [Micr16]

## **Zugriffszeiten auf die Produktiv-Datenbank**

Ein großer Nachteil der Ablage von zusätzlichen Tabellen zur Haltung von aggregierten Daten ist, dass die Produktiv-Datenbank zusätzlich belastet wird. Für das Auslesen der Daten für die Berechnung sind Performance-Einbußen im Produktivbetrieb zu vermerken. Zudem wird zusätzlicher Speicherplatz für die Daten in der Produktiv-Datenbank benötigt.

## **Data-Warehouse**

Eine weitere Möglichkeit zur Verdichtung der Daten, stellt die Verwendung eines Data-Warehouses dar. Un-

ter einem Data-Warehouse ist eine Systemlösung zu verstehen, die die unternehmensweite Versorgung der Front-End-Systeme zur Managementunterstützung mit den benötigten Informationen versorgt.

Aufgrund der Funktionen und Vorteile, stellt ein Data-Warehouse eine gute Lösung der zugrundeliegenden Problemstellung dar. Dadurch, dass die Daten für das Q-Daten-Schema bereits durch einen Parser in die Datenbank eingepflegt werden, könnte dieser auch das separate Data-Warehouse befüllen. Dadurch könnte die Last des Auslesens der Daten sowohl zur Erstbeschaffung der Daten, als auch zur Aktualisierung der Daten stark reduziert werden.

Bei Verwendung eines Data-Warehouses ist bei der Datenarchivierung darauf zu achten, dass Daten, die im Warehouse benötigt werden, auch über einen ausreichenden Zeitraum zur Verfügung stehen. Dies bedeutet, dass die benötigten Daten nur geringfügig verdichtet werden dürfen, um weiterhin wichtige Daten für die Analyse zur Verfügung zu halten.

Für die verschiedenen Standorte könnten für das Auslesen der benötigten Daten unterschiedliche Zeitpunkte gewählt werden. Das erstmalige Auslesen der Daten sollte möglichst an einem Wochenende erfolgen, um den Produktivbetrieb möglichst wenig zu beeinträchtigen.

## **Performancesteigerung bei statistischen Auswertungen**

Ein großer Vorteil dieser Variante – gegenüber den zuvor genannten – ist, dass ein Data-Warehouse so konzipiert werden kann, dass es nicht nur für einen Teil der Abfragen auf das Q-Daten-Schema, sondern für alle Abfragen eine Performancesteigerung herbeiführen kann. Durch die Auflösung der dritten Normalform und Umstrukturierung der Tabellen, können die besonders häufig benötigten Daten mit weniger *Joins* abgefragt werden.

## **Verwaltungsaufwand**

Ein Data-Warehouse einzurichten und entsprechend dem Unternehmensbedarf anzupassen, erfordert die Analyse des Gesamtsystems. Hierbei muss ein Konzept zur strukturierten und abfrageorientierten Ablage der Daten entwickelt werden. Zudem muss der komplette ETL-Prozess automatisiert werden. Dies erfordert die Betrachtung der gesamten Tabellen des MES, da zur Abfrage häufig zusätzliche Informationen zu den Elementen benötigt werden, die nicht im Q-Daten-Schema enthalten sind.

## **Kosten**

Die Kosten eines Data-Warehouses sind abhängig von der Wahl des Anbieters. Wie auch bei relationalen Datenbanken, gibt es hier deutliche Unterschiede. Je nach Aufbau des Data-Warehouses und des gewünschten Funktionsumfangs bieten sich unterschiedliche Möglichkeiten. Sowohl bei der Anschaffung der Hardware,

als auch bei der Software können die Kosten stark variieren.

### **Wiederverwendbarkeit bestehender Systeme**

Bisher bestehende Systeme müssten je nach Änderung der Struktur des Data-Warehouses stark angepasst werden. Der Zugriff per *SQL* auf die Daten wäre zwar weiterhin möglich, jedoch müssten die bisherigen Abfragen entsprechend auf die neue Struktur des Data-Warehouses angepasst werden.

### **Erweiterung des MES um Abfragemöglichkeiten des PDM**

Bei einer Erweiterung der Abfragemöglichkeiten gibt es keinerlei Einschränkungen. Bei neuen Abfragen wäre lediglich zu klären, ob diese auf dem Produktivsystem oder auf dem Data-Warehouse ausgeführt werden sollen (oder beides).

### **Herstellerunabhängigkeit**

Die Verwendung eines Data-Warehouses beschränkt sich nicht auf konkrete Anbieter. Eine Verwendung eines Produktes aus dem Hause Oracle ist naheliegend, da hierfür das entsprechende Know-how bereits vorhanden ist. Jedoch ist die Umsetzung auch in anderen Systemen möglich.

### **Zugriffszeiten auf die Produktiv-Datenbank**

Die Zugriffszeiten auf die Produktiv-Datenbank können deutlich verbessert werden, da lang andauernde Abfragen, die bislang auf dem Produktivsystem ausgeführt werden, in ein externes System ausgelagert werden können. Dadurch wird die Last auf dem Produktivsystem stark reduziert.

### **Datenarchivierung**

Die dritte Ebene des mehrschichtigen Konzeptes sieht eine langfristige digitale Archivierung vor. Hierbei sollen Daten, die nicht mehr regelmäßig im Produktivbetrieb benötigt werden, nach bestimmten Kriterien in ein externes System ausgelagert werden. Dazu wird zunächst ermittelt, was als Trennungskriterium der Daten dienen kann. Im Anschluss daran werden zwei Lösungsansätze zur Umsetzung der Datenarchivierung vorgestellt und evaluiert.

Ein klassisches Archiv beschreibt das OAIS-Referenzmodell (vgl. [CCSD12]). Das vorgestellte Prinzip eines digitalen Archivs soll als Grundlage der vorgestellten Ansätze dienen. Beide Lösungsansätze müssen die Kernkomponenten eines digitalen Archives beinhalten und die Aufgabenbereiche abdecken. Gesetzliche Vorgaben haben hierbei jedoch keine besondere Bedeutung.

Zur Archivierung digitaler Daten existieren bereits Softwarelösungen, die für verschiedene Anwendungsfälle geschaffen wurden. Hierbei ist jeweils entscheidend, in welcher Form die zu archivierenden Daten

vorliegen. Handelt es sich um Papier-Dokumente, um bereits digitalisierte Dokumente, um digital erzeugte Dateien (Email, Textdokumente, PDF...) oder strukturiert abgelegte Daten. Die meisten Systeme zur Langzeitarchivierung beziehen sich auf die Archivierung betrieblicher Daten, wie beispielsweise E-Mails und Dokumente. Dabei wird häufig besonderer Fokus auf die Erfüllung gesetzlicher Vorgaben zur Langzeitarchivierung gelegt.

Zur Archivierung von relationalen Datenbanken existiert ebenfalls bereits ein System CHRONOS [CSP 17]. Dieses wurde bereits als mögliche Lösung zur Datenarchivierung bei der Zollner Elektronik AG untersucht, jedoch wurde die Leistungsfähigkeit des Systems für den vorgegebenen Verwendungszweck als ungenügend bewertet.

### **Trennungskriterium zur Datenarchivierung**

Für eine Archivierung sind nicht nur die Relationen des Q-Daten-Schemas zu betrachten, sondern alle sich in der Masterdatenbank befindlichen Relationen.

Um eine zu frühe Auslagerung eines noch benötigten Datensatzes in das Archivsystem zu vermeiden, empfiehlt es sich, die Daten nicht rein nach zeitlichen Kriterien auszulagern. Der Auftragsstatus sollte ebenso betrachtet werden. Solange ein Auftrag nicht abgeschlossen ist, soll dieser (und alle zugehörigen Daten) nicht archiviert werden. Da die Auftragsdauer zwischen Tagen und Jahren variiert, wäre es hier kaum möglich einen sinnvollen Trennungszeitraum zu wählen. Deshalb sollte der Auftragsstatus zur Entscheidung über die Auslagerung herangezogen werden. Da es auch möglich ist, dass Datensätze nach dem Abschluss eines Auftrages erneut benötigt werden, beispielsweise wenn Fehlbuchungen passiert sind oder Kunden Produkte reklamieren, sollten die zugehörigen Daten zu einem Auftrag noch mindestens sechs Monate nach Abschluss des Auftrages im Produktivsystem gehalten werden. Somit ist sichergestellt, dass eine Rückführung der Daten aus dem Archivsystem nicht regelmäßig stattfinden muss.

### **Archiv-Datenbank per GoldenGate anbinden**

Eine Möglichkeit zur Archivierung der länger nicht benötigten Daten, stellt die Verwendung einer Archiv-Datenbank dar. Dabei handelt es sich um eine Datenbank, die den gleichen Aufbau wie die Master-Datenbank besitzt. In diese würden die zu archivierenden Daten nach dem vorher vorgestellten Prinzip ausgelagert werden. Zur Überführung der Daten könnte, wie bereits zur Synchronisierung der Datenbanken an den verschiedenen Werksstandorten mit der Master-Datenbank, *GoldenGate* von Oracle verwendet werden. Das Löschen aus der Archiv-Datenbank müsste dabei explizit unterbunden werden. Sobald die Daten im Archiv abgelegt sind, könnten die Daten aus der Produktiv-Datenbank mittels *Delete-Job* entfernt werden. Dabei kann der *Delete-Job* manuell erfolgen oder mittels *SQL*-Prozedur und entsprechenden Triggern automatisiert werden. Die Clients könnten weiterhin wie ge-

wohnt Daten abfragen. Abbildung 10 visualisiert diesen Aufbau schematisch. Sollten die benötigten Daten nicht mehr im Produktivsystem enthalten sein, könnte das Archiv zur Datenabfrage leicht herangezogen werden.

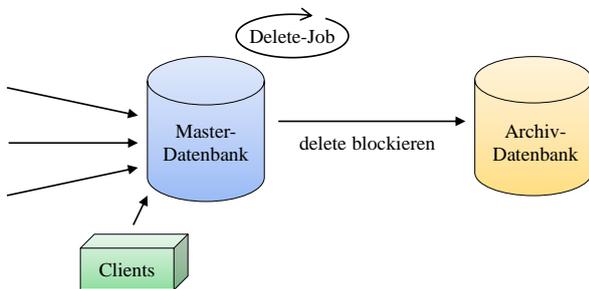


Abbildung 10: Anbindung einer Archiv-Datenbank per GoldenGate

### Rückführbarkeit bei Extraktion

Durch die Beibehaltung der Datenbank-Struktur im Archiv-System, ist der Zugriff mittels *SQL* auf das Archiv zu jeder Zeit gewährleistet. Da die archivierten Daten nicht in ein anderes Format konvertiert werden würden, müssten diese später nicht wieder zurückgeführt werden, sondern könnten wie gewohnt abgefragt werden. Bei Verwendung leistungsfähiger Hardware und Indizes (wie bereits im Produktivsystem) für die Archiv-Datenbank, ist davon auszugehen, dass diese mit einfachen Abfragen zur Wiederherstellung alter Daten keine Performanceprobleme aufweist.

### Verwaltungsaufwand

Der Verwaltungsaufwand bei der Verwendung einer Archiv-Datenbank ist relativ gering. Da die Daten in der gleichen Struktur wie bisher abgelegt werden, ist kein Konzept zur Ablage zu konzipieren, bei dem Metadaten zur Rückführung der Datensätze in das Produktivsystem berücksichtigt werden müssten. Änderungen an der Struktur der Datenbank würden mittels Synchronisierung übertragen und somit immer der aktuellen Struktur des Produktivsystems entsprechen. Lediglich für die existierenden Abfrage-Tools müsste gegebenenfalls eine Möglichkeit geschaffen werden, um auf die archivierten Daten zugreifen zu können. Dies kann im einfachsten Fall so realisiert werden, dass mittels Auswahlmenü gewählt werden kann, ob für die gewünschte Abfrage die Produktiv- oder die Archiv-Datenbank herangezogen werden soll.

### Neuanschaffungskosten und laufende Kosten

Lizenzkosten für Oracle *GoldenGate* wären nicht extra zu berücksichtigen, da das Unternehmen diese Technologie ohnehin benötigt. Jedoch würden Kosten für die Bereitstellung der entsprechenden Hardware und Lizenzkosten für eine weitere Oracle Datenbank in der Enterprise Edition anfallen. Die Erstanschaffungskosten für die Oracle Datenbank würden sich nach dem bereits vorgestellten Oracle-Lizenzmodell [Orac16] auf

$$1 \text{ Prozessor} * 47.500 \text{ US Dollar} = 47.500 \text{ US Dollar}$$

belaufen. Darüber hinaus würden jährliche Kosten für die Software-Update-Lizenzen und den Support in Höhe von etwa

$$1 \text{ Prozessor} * 10.450 \text{ US Dollar} = 10.450 \text{ US Dollar}$$

anfallen.

### Speicherplatzreduzierung im Produktivsystem

Eine Speicherplatzreduzierung im Produktivsystem wäre deutlich erkennbar. Würden nur etwa die Daten der letzten beiden Jahre der Tabelle *UNITS\_DATA* im Produktivsystem gehalten und die restlichen Daten ausgelagert, so könnten hier 54 Prozent der Daten in das Archivsystem ausgelagert werden. Ein genauer Wert kann hier nur ermittelt werden, würde die Fragmentierung mit berücksichtigt. Sobald Nutzdaten ausgelagert werden, verringert sich zudem auch die Größe der angelegten Indizes um den gleichen Faktor.

### Zugriffszeiten auf die Produktiv-Datenbank

Durch die Auslagerung der Daten in ein Archivsystem würden sich auch die Zugriffszeiten auf die Produktiv-Datenbank verringern. Durch die reduzierte Datenmenge, besonders in den häufig benötigten Tabellen, wäre eine entsprechende Performancesteigerung der Abfragen auf dem Produktivsystem zu vermerken.

### Hadoop-Cluster

Eine weitere Möglichkeit zur Datenarchivierung stellt die Verwendung eines *Hadoop*-Clusters dar. Die archivierten Daten würden dabei in einem *Hadoop*-Cluster abgelegt. Mittels *Hive* könnte dann auf die Daten über die Abfragesprache *HiveQL* zugegriffen werden. Dadurch könnten die Clients über eine standardisierte Schnittstelle Zugriff auf das Cluster erhalten. Für die Aufgabe der Extraktion und gegebenenfalls der Transformation der Daten müsste ein zusätzliches Programm geschaffen werden. Dieses wäre zudem für das Löschen der archivierten Datensätze aus dem Produktivsystem zuständig. Hierbei wäre zu berücksichtigen, dass auch die Meta-Informationen mit abgespeichert werden müssten, da die Daten in dem Cluster nicht in der bisher gewohnten relationalen Struktur abgelegt wären. Abbildung 11 visualisiert diesen Aufbau.

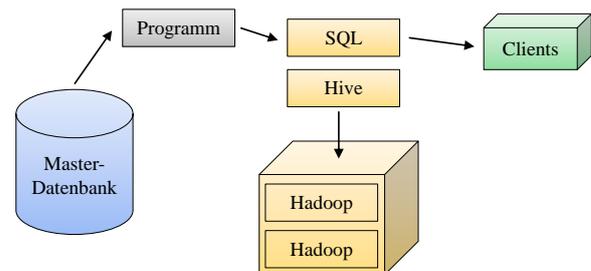


Abbildung 11: Aufbau eines Hadoop-Clusters

### **Rückführbarkeit bei Extraktion**

Die Rückführbarkeit der Daten nach der Archivierung ist aufgrund der Zugriffsmöglichkeit auf das *Hadoop*-Cluster mittels *SQL*-Syntax gegeben. Je nach Leistungsfähigkeit des Clusters sind hierfür geringe Zugriffszeiten zu erwarten.

### **Verwaltungsaufwand**

Der Verwaltungsaufwand des *Hadoop*-Clusters ist deutlich höher, als bei dem Lösungsansatz der Einführung einer Archiv-Datenbank, da hierbei die Hardware entsprechend installiert und konfiguriert werden muss.

### **Neuanschaffungskosten und laufende Kosten**

Zur Einrichtung eines Clusters wäre eine entsprechende Hardware anzuschaffen. Hierbei genügen im einfachsten Fall zwei Rechner ohne besondere Leistungsfähigkeit. Für das Framework würden keine Lizenzgebühren anfallen, da es sich dabei um eine Open-Source-Software handelt. Jedoch müsste ein Programm geschaffen werden, das die benötigte Funktionalität zum Auslagern und Rückführen der Daten ermöglicht. Dieses könnte extern oder intern entwickelt werden. Hierfür würden entsprechend dem angesetzten Aufwand zusätzliche Kosten entstehen. Für den Betrieb und die Wartung der Systeme und des Programms würden je nach Bedarf weitere Kosten anfallen.

### **Speicherplatzreduzierung im Produktivsystem**

Die Speicherplatzeinsparung bei der Verwendung eines *Hadoop*-Cluster entspricht der gleichen Einsparung wie bei dem Einsatz einer zusätzlichen Archiv-Datenbank, da die Daten nach gleichen Kriterien ausgelagert werden würden.

### **Zugriffszeiten auf die Produktiv-Datenbank**

Die Verbesserung der Zugriffszeiten auf die Produktiv-Datenbank entspricht der gleichen Steigerung wie bei der Verwendung einer Archiv-Datenbank, da eine Verbesserung der Zugriffszeiten lediglich auf die Reduzierung der Anzahl der zu lesenden Datensätze zurückzuführen ist und diese bei beiden Systemen gleich hoch wäre.

### **Vorschlag zur Realisierung**

Nachdem im letzten Kapitel mehrere Lösungsansätze vorgestellt und evaluiert wurden, wird nun eine Empfehlung zur Realisierung des mehrschichtigen Konzeptes bei der Zollner Elektronik AG gegeben.

Aufgrund der hohen Kosten bei der Verwendung der Option der Partitionierung bei einer Oracle Datenbank und der Tatsache, dass eine Datenverdichtung mittels Materialisierter Sicht das Produktivsystem zu sehr belastet, empfiehlt sich die Verwendung eines Data-Warehouses zur Datenverdichtung. Hierbei ist es nahelegend ein System von Oracle zu verwenden, jedoch

könnte hierzu auch jede andere Datenbank, die diese Funktionalität bietet, verwendet werden.

Für die Datenarchivierung wurden die Möglichkeiten der Anbindung einer zusätzlichen Archiv-Datenbank und die Verwendung eines *Hadoop*-Clusters untersucht. Beide Ansätze erfüllen die gestellten Anforderungen. Die Ansätze unterscheiden sich lediglich in den Anforderungen *Rückführbarkeit bei Extraktion*, *Verwaltungsaufwand* und *Kosten*. Somit ist bei der tatsächlichen Umsetzung zu entscheiden, ob die Anforderung nach weniger Verwaltungsaufwand und einer einfacheren Rückführbarkeit der Daten oder die Kosten höher gewichtet werden sollen. Ist ein geringer Verwaltungsaufwand und eine einfache Rückführbarkeit der Daten gewünscht, empfiehlt sich die Anbindung einer zusätzlichen Archiv-Datenbank per *GoldenGate*. Liegt jedoch der Fokus auf einer kostengünstigeren und skalierbaren Lösung, empfiehlt sich die Umsetzung eines *Hadoop*-Clusters.

### **Zusammenfassung und Ausblick**

In diesem Kapitel werden alle wichtigen Inhalte der Arbeit kurz zusammengefasst. Zudem wird erklärt, warum das Konzept nur schwer auf andere Sachverhalte übertragbar ist.

Das Ziel war die Konzeptionierung einer mehrschichten Haltung für die Mess- und Qualitätsdaten der Zollner Elektronik AG. Hierzu wurde zunächst die IT-Infrastruktur des Unternehmens dargelegt und die Ausgangssituation detailliert beschrieben. Anschließend wurde mit Hilfe einer Anforderungsanalyse genau ermittelt, welche Anforderungen von Seiten des Unternehmens an das Konzept gestellt werden. Hierbei wurde das Konzept in die drei Ebenen *Produktivdaten*, *Datenverdichtung* und *Datenarchivierung* unterteilt, um die Problemstellung möglichst genau untersuchen zu können.

Im Kapitel Konzept-Entwurf wurden mehrere Lösungsansätze zur Realisierung der Schichten *Datenverdichtung* und *Datenarchivierung* vorgestellt und anhand der ermittelten Anforderungen evaluiert. Im Anschluss daran, wurde eine Empfehlung zur konkreten Umsetzung des Konzeptes gegeben.

Abschließend lässt sich sagen, dass für jede der dargelegten Schichten die Möglichkeit einer Umsetzung gegeben ist. Dabei ist anzumerken, dass das Konzept auf den sehr konkreten Anforderungen der Zollner Elektronik AG basiert und damit nur schwer mit anderen allgemeingültigen Konzepten vergleichbar beziehungsweise auf andere Datenschemen und Sachverhalte anwendbar ist. Durch Abänderung der Anforderungen oder deren Prioritäten, kann sich bei gleicher Fragestellung eine andere Lösung als besser geeignet herausstellen.

Würden bei der Datenverdichtung nicht nur die Mess- und Qualitätsdaten betrachtet werden, sondern die komplette Datenbasis des Zollner-MES, so wäre möglicherweise die Verwendung von SAP BW HANA als möglicher Lösungsansatz zu untersuchen.

## Literatur

- [Brot03] *Brot Tatjana*: Multimedia-Seminar - Partitionierungstechniken in Datenbanksystemen. 03.03.2003, <https://homepages.thm.de/~hg10013/Lehre/MMS/SS02/Brot/text.htm>. Abruf am: 2017-01-04.
- [CCSD12] CCSDS: Reference Model for an Open Archival Information System (OAIS).
- [CSP 17] CSP GmbH & Co. KG: CHRONOS Datenbankarchivierung - CSP Software. 2017, <https://www.csp-sw.de/de/produkte/chronos-datenbankarchivierung/>. Abruf am: 2017-01-23.
- [Micr16] Microsoft: Create Indexed Views. 2016, <https://msdn.microsoft.com/en-us/library/ms191432.aspx>. Abruf am: 2017-02-09.
- [Orac16] Oracle Corporation: Oracle Technology Global Price List. 2016, <http://www.oracle.com/us/corporate/pricing/technology-price-list-070617.pdf>. Abruf am: 2017-01-01.
- [Schi14] *Schicker Edwin*: Datenbanken und SQL. Eine praxisorientierte Einführung mit Anwendungen in Oracle, SQL Server und MySQL. Springer Vieweg, Wiesbaden, 2014.
- [Zoll16] Zollner Elektronik AG: EMS Elektronikdienstleister Zollner Elektronik AG. 2016, <https://www.zollner.de/>. Abruf am: 2017-01-01.
- [Zühl11] *Zühlke Karin*: MES-Plattform intraFactory von Kratzer ist kein »Serienprodukt« mehr: MES - (k)ein Schreckgespenst? 2011, <http://www.elektroniknet.de/markttechnik/elektronikfertigung/mes-k-ein-schreckgespenst-81209.html>. Abruf am: 2017-01-01.