

# **Revisionsicherheit & Datenbanken – wie setze ich das um?**

**Heike Johannes  
CSP GmbH & Co. KG  
Großköllnbach**

**Reimund Willig  
EMC Deutschland GmbH  
München**

## **Schlüsselworte:**

Compliance, Revisionsicherheit, Datenbankarchivierung, Content-Addressed Storage, Chronos, Centera

## **Einleitung**

Alle international tätigen Unternehmen sind in zunehmendem Maße gefordert, Compliance-Vorschriften umzusetzen. Dabei spielt die Revisionsicherheit für Datenbanken eine große Rolle. Dieser Vortrag erläutert einen gesamtheitlichen Ansatz, wie in diesem Zusammenhang Revisionsicherheit umgesetzt werden kann.

## **Gründe für Revisionsicherheit bei Datenbanken**

Aktuellen Studien zufolge wird sich die Masse der elektronisch gespeicherten Informationen in den nächsten Jahren explosionsartig vergrößern. In allen Branchen wächst mit der verfügbaren Rechenleistung auch der benötigte Speicherplatz für Daten. Dadurch steigt auch der Bedarf zur Langzeitspeicherung von Datenbankinhalten, denn eine Vielzahl dieser Daten muss aus gesetzlichen Gründen oder aufgrund der Geschäftsprozesse längerfristig gespeichert werden.

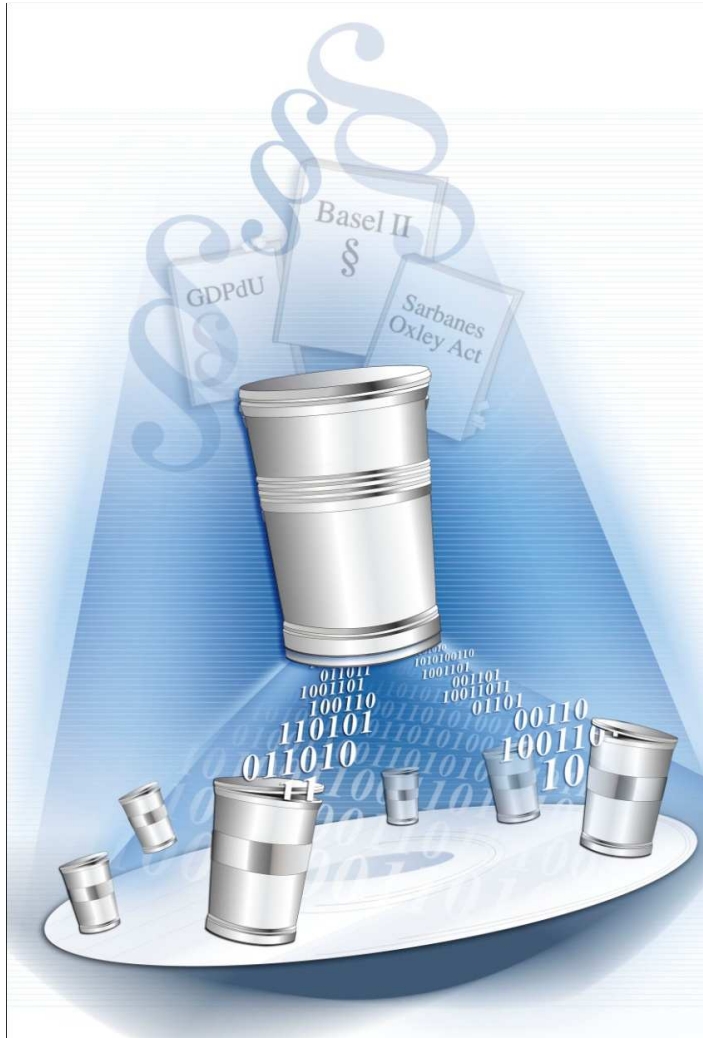


Abbildung 1: Gesetzliche Auflagen bei Datenbanken

Gleichzeitig steigen auch die Sicherheitsanforderungen an elektronische Speichersysteme, denn immer öfter werden Daten nur noch elektronisch gespeichert und über Jahrzehnte aufbewahrt. Papieroriginale treten zunehmend in den Hintergrund. Hinzu kommt, dass für die Speicherung elektronischer Daten landesspezifische gesetzliche Regelungen und Normen zu berücksichtigen sind. In Deutschland beispielsweise existieren u.a. folgende Anforderungen:

- Handels- und Steuerrecht
- Produkt- und Organisationshaftung
- Bundesdatenschutzgesetz (BDSG)
- Grundsätze zum Datenzugriff und zur Prüfbarkeit digitaler Unterlagen (GDPdU)
- Sozialgesetzbuch (SGB)
- Signaturgesetz (SigG)

### **Eine Datenbank allein ermöglicht keine Revisionsicherheit**

Zum Einhalten der oben genannten Gesetze ist die Revisionsicherheit oberstes Ziel. Hierzu gehört die Unveränderbarkeit der Datenbankinhalte. Datenbanken sind allerdings darauf ausgelegt, dass jederzeit ein Update erfolgen kann. Somit widerspricht die Revisionsicherheit dem eigentlichen Datenbankziel.

Das Read-Only-Setzen eines Tablespaces ist in der Praxis meist schwer durchzusetzen, da nicht alle darin enthaltenen Daten unveränderbar sein sollen.

Zudem fehlt die Authentizität der Daten: in der Standard-Version einer Datenbank ist die Authentizität der Informationen nicht gegeben. Eine Authentizität ist beispielsweise durch das Verwenden einer digitalen Signatur möglich.

### **Aspekte einer Datenbankarchivierung**

Eine Datenbankarchivierung übernimmt folgende Aufgaben:

- ✂ Identifizieren inaktiver Informationen aufgrund von Regeln
- ✂ Umwandeln in ein offenes, langzeitsicheres Format
- ✂ Auslagern der Daten in ein Archiv
- ✂ Bereitstellen der archivierten Informationen (z. B. per SQL-Abfrage)
- ✂ Löschen der Daten nach Ablauf der Aufbewahrungsfrist

Sobald eine Datenbankarchivierung zur Aufbewahrung bestimmte Daten aus einer Produktivumgebung auslagert, ermöglicht sie damit:

- ✂ Revisionsicherheit & Langzeitlesbarkeit
- ✂ geringeres Datenbankvolumen (Daten und Indizes)
  - kleinere Backup-Umfänge
  - kürzere Recovery-Vorgänge
  - leichtere Migration
  - niedrigere Speicherkosten
  - bessere Performance
  - geringere Betriebskosten

Eine Datenbankarchivierung sollte dabei folgende Anforderungen erfüllen:

- ✂ Offenes Archivformat:  
Meta- und Nutzdaten sollten in einem offenen Format im gleichen Dateisystem abgelegt werden, um einen konsistenten Container zu erhalten. Dabei bietet es sich an, bei Nutzdaten auf Textdateien mit wählbarem Characterset zurückzugreifen. Für Metadaten der archivierten Umfänge (Tabellennamen, Spaltennamen, Datentypen, Indexinformationen, Constraints, ...) könnten XML-Dateien genutzt werden.
- ✂ Inkrementelle Archivierung:  
Das Archiv sollte inkrementell aufgebaut sein und gleichzeitig die relationalen Abhängigkeiten zwischen den Tabellen beachten. Dann entstehen in sich geschlossene, konsistente Archivpakete.
- ✂ Behandeln von Strukturveränderungen:  
Strukturveränderungen stehen über den Archivierungszeitraum hinweg beinahe auf der Tagesordnung. Deshalb sollte eine Archivierungssoftware diese erkennen und automatisiert in die Konfiguration des Archivs einarbeiten. Dann verursachen auch umbenannte Datenbankobjekte (z. B. Spalten, Tabellen) oder neue Spalten mit Constraints keinen Bruch im Archivzugriff. Ein wichtiger Aspekt an der Stelle ist, dass niemals die Archivdateien bei Schemaänderungen migriert werden müssen. Vielmehr sollte eine abstrakte Zugriffsschicht die Daten virtuell an das aktuelle Schema anpassen, auf die gerade zugegriffen werden soll. Dies verkürzt bei Applikationsupdates drastisch die Gesamtlaufzeit, da nur die online gehaltene Datenmenge migriert werden muss.
- ✂ Migrationen bei Datenbankversionen:

Einen ähnlichen Effekt erzielt man bei Migrationen zwischen Datenbankherstellern bzw. innerhalb eines Upgradepfades eines Herstellers. Durch das abgespeicherte offene Archivformat muss hier wiederum nur der online gehaltene Datenumfang migriert werden. Die Archivdaten bleiben unverändert und werden erst bei einem eventuellen Rückspielvorgang an die neue Datenbanksoftware angepasst.

Allerdings ist die Datenbankarchivierung allein in der Regel nicht revisionssicher: jederzeit könnten Benutzer auf die generierten Dateien zugreifen und diese verändern oder sogar löschen. Deshalb ist der Einsatz eines dedizierten, hardware-basierten Archivsystems unumgänglich. Dieses muss die Funktionalitäten bieten, die für eine revisionssichere Speicherung notwendig sind. Dazu gehören gesicherte Verfahren zur Herstellung und Nachweis der Integrität (inhaltliche Echtheit) und Unveränderbarkeit der gespeicherten Information sowie das zertifizierte Löschen. Desweiteren ist entscheidend, dass das Archivsystem über eine offene Softwareschnittstelle verfügt, die eine Integration in verschiedene Applikationsumgebungen erlaubt. Somit wird auch der Einsatz des Systems als sogenanntes 'Single Archive Repository' ermöglicht.

### **Plattenbasiertes Archivsystem**

Die klassischen Konzepte zu Langzeitspeicherung und Auslagerung von Daten (Band, DVD, CD) werden heutzutage abgelöst durch neuartige Speichertechnologien. Diese sind in der Lage, große Datenmengen im Online-Zugriff zu halten. Gleichzeitig erfüllen sie die oben erwähnten Grundanforderungen nach Authentizität und Unveränderbarkeit der Information sowie das zertifizierte Löschen.

Aktuelle Magnetplatten-basierte Archivsysteme sind speziell für die Speicherung von Daten entwickelt, die selten oder gar nicht mehr verändert werden. Für den Datenzugriff wird hier die sogenannte „Content Addressed Storage“-Speichertechnologie (CAS) verwendet.

Der Datenzugriff erfolgt ausschließlich über Anwendungen, die mittels einer API (Application Program Interface) in die Lage versetzt werden, mit dem System Daten auszutauschen. Durch die Verwendung von Festplatten ist ein Online-Zugriff auf die Daten möglich.

Bei der „Content Addressed Storage“-Speichertechnologie werden folgende Schritte durchgeführt:

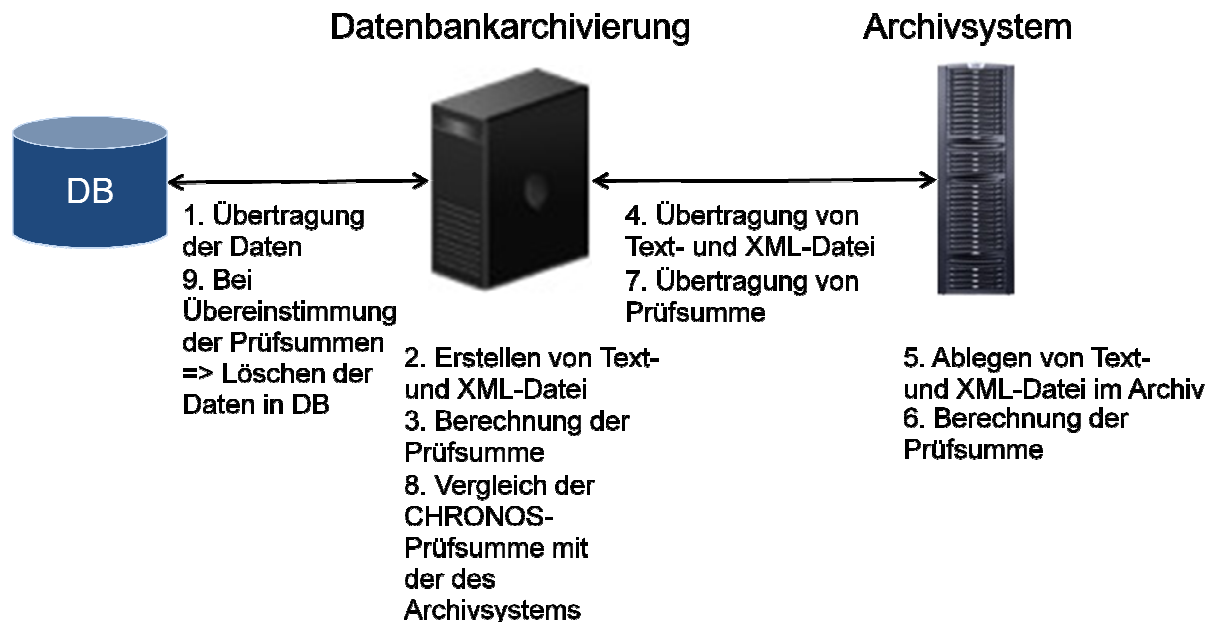
- ✎ **Erstes Abspeichern:**  
Beim ersten Abspeichern eines Datenobjekts wird der Anwendung über die API des Archivsystems eine weltweit eindeutige Identifikation (Content Address – CA), vergleichbar einem Fingerabdruck, zurückgegeben. Diese CA wird aus dem binären Inhalt des zu speichernden Objekts errechnet. Gleichzeitig werden die das Objekt betreffenden Metadaten in einem so genannten CDF (Content Descriptor File) gespeichert, welches als XML-basiertes File von der Datenbankarchivierung mit beliebigen Merkmalen (z. B. Aufbewahrungsfristen) ergänzt werden kann.
- ✎ **Datenzugriff:**  
Zum Wiederauffinden der Daten übergibt die Datenbankarchivierung die CA über die API an das Archivsystem. Die CA ist die einzige verfügbare Referenz für die Daten. Es werden weder Verzeichnisse, Pfadnamen noch URLs verwendet. Der physikalische Speicherplatz der Daten ist für die Anwendung nicht transparent.
- ✎ **Versionsintegrität:**  
Wird ein bereits gespeichertes Objekt gelesen, geändert und erneut abgespeichert, so behandelt das Archivsystem dieses Objekt wie ein neues Objekt. Da bei der Abspeicherung eine vom Originalobjekt unterschiedliche CA erzeugt wird, erzeugt das System ein weiteres CDF. Das Originalobjekt bleibt unverändert und über seine ursprüngliche Content-Adresse zugreifbar. Diese „Write Once Read Many“ (WORM) Funktionalität ermöglicht eine Versionsintegrität.

Für Anwendungsentwickler und Speichermanager ist der Speicherort der Daten transparent. Die Speicher- und Performanceoptimierung sowie die redundante Speicherung werden vom Archivsystem übernommen. Lediglich die Content-Adresse ist die global einmalige Identifikation, die es ermöglicht, Datenobjekte wieder aufzufinden, unabhängig vom Ort der Speicherung und der Anwendung, die die Objekte erzeugt hat.

Verschiedene Applikationen können zudem unbeeinflusst voneinander ihre Daten auf dem Archivsystem ablegen. Durch die Bildung von so genannten virtuellen Pools schreibt die Datenbankarchivierung für jede einzelne Applikation die Daten in eigene, logische Bereiche, die auch einzeln auf andere Archivsysteme repliziert werden können.

### Gesamtheitlicher Ansatz

Durch die Kombination der Produkte von CSP und EMC steht Kunden eine aufeinander abgestimmte Lösung zur Verfügung, die den schnellen Zugriff auf archivierte Daten bei gleichzeitiger Konformität mit Compliance-Anforderungen ermöglicht.



Die Lösung bildet eine perfekte Symbiose bei der Archivierung von inaktiven Datenbankbeständen (vgl. Abbildung 2): die Datenbankarchivierung löst die Daten aus einer relationalen Datenbank heraus und wandelt sie in ein langfristig lesbares Format um. Diese Daten überträgt sie dann auf das Archivsystem. Dessen Aufgabe besteht darin, als WORM-Storage-System die Revisionsicherheit zu gewährleisten. So können die Daten später nicht mehr verändert werden. Die integrierte Daten-Replikation des Archivsystems ermöglicht sicher, dass die Daten immer vorhanden sind und damit lesbar bleiben. Einstellbare Retention-Zeiten ermöglichen zusätzlich das Löschen der Daten zu einem definierten Zeitpunkt.

Abbildung 2: Ablauf eines Archivierungsvorgangs

**Kontaktadresse:**

**Heike Johannes**

CSP GmbH & Co. KG  
Herrenäckerstr. 11  
D-94431 Großköllnbach

Telefon: +49 (0) 9953-3006 22  
Fax: +49 (0) 9953-3006 50  
E-Mail [heike.johannes@csp-sw.de](mailto:heike.johannes@csp-sw.de)  
Internet: [www.datenbankarchivierung.de](http://www.datenbankarchivierung.de)

**Reimund Willig**

EMC Deutschland GmbH  
Osterfeldstr. 84  
D-85737 Ismaning

Telefon: +49 (0) 89-930 910  
Fax:  
E-Mail [reimund.willig@emc.com](mailto:reimund.willig@emc.com)  
Internet: [www.emc.com](http://www.emc.com)