Datenauslagerung aus Datenbanken – Fallstricke aus der Praxis

Mario Täuber
CSP GmbH & Co KG
Großköllnbach

Schlüsselworte
Datenauslagerung, Datenbankarchivierung, Extraktion, inaktive Daten.

Einleitung

Die Datenhaltung in einer Oracle Datenbank bietet viele Vorzüge, wie hohe Zugriffsgeschwindigkeit, hohe Verfügbarkeit, Dauerhaftigkeit und Transaktionssicherheit. Alle diese Eigenschaften werden naturgemäß durch einen organisatorisch sehr hohen Aufwand und folglich auch mit sehr hohen Kosten erkauft. Benötigt man noch dazu eine Schattendatenbank um sehr hohe Verfügbarkeit herstellen zu können, verdoppelt sich dieser Aufwand nahezu. Für hochaktuelle und häufig benötigte Daten in den Unternehmen ist dies natürlich ohne Frage gerechtfertigt. Gilt dies jedoch auch für historische Daten, die in der Realität enorm selten bis nie abgerufen werden?

Diese Frage haben sich auch bereits viele Unternehmen gestellt und mit "nein" beantwortet. In der Folge davon wurden Projekte gestartet, die die Auslagerung dieser Altdaten aus der teuren Onlinedatenbank realisieren sollten. Häufig wurde dabei zur Umsetzung auf Praktikanten oder Diplomanten zurückgegriffen. Die technische Realisierung spannt sich dabei von der Verwendung von Oracle Bordmitteln über bereits sehr ausgefuchste Scriptinglösungen bis hin zu kleinen selbst geschriebenen Archivierungsprogrammen. Welche Fallstricke sich daraus im täglichen Betrieb oder im Zugriffsfall auf diese ausgelagerten Daten ergeben können, wird in dieser Ausarbeitung gezeigt.

Auslagerung mittels Oracle Export

Um die eingangs erwähnte Datenauslagerung aus der Produktionsdatenbank erreichen zu können, wählen viele Anwender den Oracle Export als Hilfsmittel aus. Dabei werden die größte Tabelle oder die größten Tabellen mit Hilfe des Export Tools abgezogen und die online nicht mehr benötigten Datensätze aus der Datenbank gelöscht. Nach einer Reorganisation kann die Datenbank wieder weiterwachsen. Dieses Vorgehen wird in zyklischen Zeitabständen wiederholt, da die Größe der Datenbank dabei immer wieder weiter steigt. Eine Variation davon wäre das Kopieren der auszulagernden Daten via „insert into … select …“ in eine Zwischentabelle, um nur die Altdaten exportieren zu können.

Als Resultat dieses Vorgehens entstehen nun mehrere Schwierigkeiten. Zunächst ist das Löschen der so exportierten Datenumfänge nicht ganz trivial, sobald relationale Abhängigkeiten zwischen den exportierten Tabellen existieren. Dabei muss penibel darauf geachtet werden, dass die Constraints mit Cascade Optionen korrekt formuliert worden sind. Sollte dies nicht der Fall sein, so müssen abhängige Datensätze per Hand (alternativ per Skript) aus der Datenbank gelöscht werden. Häufig sind Cascade Constraints aufgrund der fachlichen Logik auch gar nicht gewünscht. Somit muss ein sehr hoher Aufwand in das korrekte Löschen der extrahierten Umfänge gesteckt werden. In so einem Szenario beschränken sich diese Lösungen meist auf den Export einer Tabelle. Auch das vollständige exportieren von relational Abhängigen Datensätzen ist extrem aufwändig und fehleranfällig.

Da der Datenzugriff bei diesem Szenario mit recht viel Aufwand verbunden ist, und auch sehr viel Zeit beansprucht, ist dieses Vorgehen nur bei Datenauslagerungsprojekten anzutreffen, bei denen sehr seltener bis kein Zugriff auf die ausgelagerten Daten erwartet wird. Dieses Restrisiko kann jedoch jederzeit zuschlagen. Sei es in Form eines Wirtschaftsprüfers, der nun doch plötzlich Altdaten prüfen will, oder auch in Form eines Gerichtsprozesses, bei dem wichtige Daten zur Entlastung des Unternehmens benötigt werden. In beiden Fällen entsteht sehr schneller Handlungsbedarf zur Wiederherstellung der Daten. Damit dies entsprechend zügig und sicher gelingt, muss die Wiederherstellbarkeit der Export Dateien zyklisch getestet werden. Dies stellt an sich schon einen schnell wachsenden, zyklischen manuellen Aufwand dar. Aus diesem Grund werden Prüfungen der Lesbarkeit der Dumpfiles häufig unterlassen. Bei diversen Prüfungen sind die Prüfer häufig zufrieden, wenn die entsprechenden Dumpfiles gezeigt werden, ein Import wird meist nicht gefordert. Benötigt aber das Unternehmen selbst schnell Zugriff auf die Dateninhalte wird festgestellt, dass dies nur mit sehr hohem Aufwand möglich ist.

Nehmen wir z.B. an, dass diese Auslagerungsstrategie auf monatlicher Basis ausgeführt wird und dies seit fünf Jahren geschieht. Somit liegen die Altdaten in 60 Exportfiles vor. Häufig werden in Gerichtsprozessen Daten über einen langen Zeitraum gefordert, beispielsweise um beweisen zu können, dass in der Produktion kein Serienfehler enthalten war. Dadurch kann es nun notwendig werden, dass alle Exportfiles durchsucht werden müssen. Dazu müssen jetzt jedoch alle Exportfiles wieder importiert werden und die geforderten Datensätze extrahiert werden. Meist wird erst zu diesem Zeitpunkt festgestellt, dass die meisten Exportfiles in unterschiedlichen Datenbankschemata vorliegen. Die Anwendung und somit das Datenbankschema wurde ja über die Dauer der Archivierung hinweg mehrmals mehr oder weniger stark geändert. Diese Migrationsschritte müssen nun bei der schrittweisen Wiederherstellung nachvollzogen werden, um alle relevanten Daten in das korrekte Schema zu transferieren. Dadurch kann leicht eine Projektlaufzeit der Wiederherstellung im Monatsbereich erreicht werden. Bei Wirtschaftsprüfungen und Gerichtsprozessen werden dabei dann meist sehr hohe Strafzahlungen fällig. Zusätzlich bindet man natürlich während dieser Projektlaufzeit Personal, welches eigentlich für andere Themen zur Verfügung stehen sollte.

Auslagerung mittels Spool Files

Auf ähnliche Probleme stößt man, wenn die Datenauslagerung mittels Spool Files im SQL Plus Befehlszeilentool von Oracle durchgeführt wird. Auch hier werden typischerweise in zyklischen Abständen via SQL Statements bestimmte Datenbereiche aus Tabellen extrahiert und anschließend gelöscht. Hier kann jedoch schon der Vorteil genutzt werden, dass nicht jedes Mal die gesamte Tabelle abgezogen werden muss, sondern im Selektionsteil des SQL Statements der inaktive Teil der Daten definiert werden kann und dann nur dieser Datenumfang in das Spoolfile geschrieben wird.

Ein weiterer Vorteil bei der Extraktion via Spoolfile ist das Format des erzeugten Files. Im Gegensatz zum proprietären Format des Dumpfiles des Oracle Exports ist das Format bei Spoolfiles offen lesbar. Dies hat den Vorteil, dass die Lesbarkeit der Dateien nicht explizit zyklisch geprüft werden muss, sondern es ausreicht, diese Dateien auf einen revisionssicheren Speicher abzulegen, der die Validität der Datei an sich garantiert. Die Wiederherstellung der Daten kann relativ einfach mit dem SQL Loader erfolgen. Bei einfachen Szenarien, wenn beispielsweise nur eine Tabelle ausgelagert wurde, kann auch die Datenmenge für den Load-Vorgang stark reduziert werden, in dem die einzelnen Zeilen mit Betriebssystemmittlen aus dem Datenfile extrahiert werden.

Ein großes Problem im Handling entsteht jedoch, falls Daten aus mehreren zusammenhängenden Tabellen extrahiert werden sollen. Die relationalen Abhängigkeiten können in diesem Fall nicht einfach berücksichtigt werden. Weiterhin besteht dasselbe Problem beim Datenzugriff, falls wie im

vorherigen Absatz beschrieben, Daten aus einem größeren Zeitbereich benötigt werden, oder falls nicht bekannt ist, in welchem Zeitbereich nach den benötigten Informationen zu suchen ist.

Auslagerung mittels Scripting

Aufgrund der Nachteile der oben genannten Möglichkeiten, Daten aus einer Oracle Datenbank auszulagern, wurden in vielen Unternehmen bereits Anstrengungen unternommen, um den Datenzugriff zu vereinfachen. Dabei werden beispielsweise direkt insert Statements in Textdateien abgelegt, um den Importvorgang zu vereinfachen. Basis dieser Auslagerungstechnologie ist letztendlich meist auch das Oracle Spooling, jedoch werden um die eigentlichen Aufrufe mehr oder weniger komplexe Skripte erzeugt.

Das hier beschriebene Vorgehen sorgt jedoch im Speicherplatz für einen relativ großen Overhead durch die häufigen Wiederholungen. Eine Optimierungsmöglichkeit ist es nun, den ersten Teil des insert Statements nur einmal am Anfang der Datendatei abzulegen. Nur der value-Teil des insert Statements wird pro Zeile gespeichert. Dies sorgt jedoch nun wiederum für eine hohe Komplexität beim Datenzugriff, da die beiden Teile des Statements zusammengefügt werden müssen. Analog diesem Vorgehensmuster wurden von uns verschiedene Optimierungen in der Praxis angetroffen, jedoch immer mit dem gravierenden Nachteil, dass der Datenzugriff letztendlich wieder komplexer wurde, je optimierter die Datenhaltung gestaltet wurde.

Auslagerung mittels PL/SQL

Einfacher bzgl. des Datenzugriffs ist die Auslagerung von inaktiven Daten via datenbankinternen Mechanismen, wie beispielsweise PL/SQL Prozeduren. Dabei werden die auszulagernden Daten in einen zweiten User auf einem anderen Tablespace auf günstigeren Speichermedien abgelegt. Der Datenzugriff ist bei dieser Lösung relativ einfach, da kein Technologiebruch stattfindet. Man kann mit denselben Mechansimen, häufig sogar direkt mit der Originalapplikation auf die ausgelagerten Daten zugreifen.

Mit dieser Lösung ist jedoch das eigentliche Problem, welches mit der Datenauslagerung gelöst werden sollte, nur verlagert. Die Archivdatenbank muss genauso wie die produktive Datenbank einem Backup unterzogen werden und bei Migrationen muss diese Datenbank ebenso eins zu eins behandelt werden. Somit müssen weiterhin Daten, die eigentlich nicht mehr angefragt werden, im normalen Betriebszyklus einer Datenbank gemanagt werden. Zusätzlich findet nun das enorme Datenwachstum auf dem Archivbereich der Datenbank statt, so dass dieser über die Zeit die in diesem Artikel diskutierten Aktionen erfordert.

In unseren Praxiseinsätzen ist dies ein sehr häufiges Ausgangsszenario, bei dem dann Altdaten aus der Archivierungsdatenbank ausgelagert werden müssen. Nach den negativen Erfahrungen mit der ersten Stufe der Auslagerung wird dann natürlich nicht mehr auf dasselbe Mittel zurückgegriffen, sondern eine Standardlösung eingesetzt.

Auslagerung mittels einer Standardlösung

Häufig steigt der Aufwand des Datenzugriffs mit dem Wechsel des für die Auslagerung verantwortlichen Mitarbeiters enorm. Die hier beschriebenen Lösungen sind aufgrund von geringen Budgets und Zeitknappheit meist schlecht dokumentiert. Viele derartige Projekte werden nach dem Motto „Hauptsache es läuft“ umgesetzt. Dabei sammelt sich bereits über einen sehr kurzen Zeitraum ein gigantischer Datenberg in den ausgelagerten Dateien an. Im Informationszeitalter sind dies meist mit die wichtigsten Assets einer Firma. Ohne Zugriff auf diese Informationen können dem Unternehmen hohe Schäden entstehen. Daher entschließen sich viele Unternehmen, diese

Datenauslagerung und Datenhaltung über eine Standardlösung abzubilden. Dies bietet neben den klassischen Vorteilen einer Standardlösung weitere Vorteile, die gezielt bei der Bewertung von Standardlösungen zur Datenauslagerung abgefragt werden sollten:

Das Dateiformat ist Datenbankherstellerunabhängig und offen verfügbar. Daten können zur Not auch immer noch ohne ein Softwareprodukt interpretiert und ausgewertet werden. Dies ist bei Aufbewahrungsfristen von 10, 15 oder 30 Jahren und mehr immens wichtig, um nicht mit zyklischen Lesbarkeitstests wie oben beschrieben belastet zu werden.

Die Erstellung der Archivdateien sollte echt inkrementell in kleinen Stufen erfolgen können, um kurze Laufzeiten der einzelnen Jobs und geringe Auswirkungen auf den Produktivbetrieb der Datenbank und Applikation zu erzeugen. Dabei werden auch relationale Abhängigkeiten zwischen Tabellen beachtet und in die Auslagerungsdateien übertragen. Somit liegen in den einzelnen Dateipaketen alle zu einem Geschäftsobjekt gehörigen Datensätze. Bei der Ablage auf Band ist dies ein unverzichtbarer Vorteil, da die Daten somit physikalisch beisammen liegen und ohne langes Positionieren gelesen werden können.

Um die Daten speichereffizient ablegen zu können, müssen die Auslagerungsdateien komprimiert abgelegt werden. Dies wird meist auch bei den oben beschriebenen Auslagerungsmechanismen gemacht. Dabei ist die Komprimierung beim Lesen der Daten dann meist ein Nachteil, da zunächst ein sehr langer Entpackvorgang durchgeführt werden muss. Bei der hohen Inkrementalität der Ablage der Daten in einer Standardlösung (meist auf täglicher Basis) entfällt dieser lange Entpackvorgang. Bei Anfragen auf die ausgelagerten Daten müssen nur die jeweils aktuell für die Anfrage benötigten Datendateien entpackt werden. Die einzelnen Datendateien sind durch die hohe Inkrementalität sehr klein, dadurch müssen für die Beantwortung einer Anfrage auch nur sehr wenige Datenmengen eingelesen und entpackt werden.

Weiterer Vorteil von Standardlösungen ist die vorhandene Anbindung an Jobsteuerungstools und Monitoringtools. Diese sind im Datenbankumfeld naturgemäß besonders wichtig, da hohe Verfügbarkeit und geringer Wartungsaufwand sehr wichtige Faktoren sind. Durch die Verwendung von existierenden Tools kann gewährleistet werden, dass der Betrieb der neuen, ergänzenden Datenhaltungskomponente unveränderten mit den bisherigen Tools und Mechanismen abgebildet werden kann.

Auch die oben beschriebene Thematik der Veränderung von Datenbankstrukturen zwischen verschiedenen Auslagerungsvorgängen muss eine Standardlösung bereits behandeln. Dies ist so gelöst, dass Altdaten bei neuen Schemaversionen im Archiv unverändert beim Lesen ‚on the fly' in die aktuelle Schemaversion transferiert werden. Dadurch entsteht der Vorteil, dass bei Applikationsupdates inaktive Datensätze nicht geupdatet werden müssen. Dies bringt Zeit- und Kostenvorteile beim Update oder der Migration von Applikationen.

Als letzten Vorteil einer Standardlösung möchte ich hier die integrierte Überprüfung der Korrektheit und Vollständigkeit der Auslagerung anmerken. Durch Integration von Mechanismen zur digitalen Signatur und Erstellung sowie Prüfung von Checksummen je Datensatz kann die Integrität der Datendateien sichergestellt werden. Des Weiteren kann über die enge Integration mit zertifizierten, revisionssichern Storagesystemen auch sichergestellt werden, dass Datensätze aus der Datenbank erst entfernt werden, wenn die korrekte und dauerhafte Ablage sichergestellt ist. Auch dies ist bei den eingangs beschriebenen Mechanismen nicht immer gewährleistet.

Fazit

Aus der hohen Anzahl an verschiedenen Möglichkeiten, inaktive Daten aus relationalen Datenbanken auszulagern, ist abzuleiten, dass dies ein in der Praxis sehr relevantes Thema ist. Nahezu jede IT Abteilung in großen Unternehmen hat sich dazu schon Gedanken gemacht, bzw. eine der oben aufgezeigten Mechanismen umgesetzt. Diese Lösungen haben wir in der Praxis angetroffen und mit den Verantwortlichen die unterschiedlichen Nachteile diskutiert. Meist geht dann der Weg mehr oder weniger schnell in Richtung Standardprodukt. Die Vorteile dieses Wegs wurden im Artikel näher erläutert. Zu erwarten ist, dass auch weiterhin Unternehmen zunächst eine einfache, selbst gestrickte Möglichkeit der Datenauslagerung umsetzen werden. Diese ist schnell und kostengünstig umzusetzen. Jedoch nach einigen Jahren des Praxiseinsatzes werden die oben gezeigten Nachteile zum Vorschein treten. Spätestens wenn zum ersten Mal nach der Datenauslagerung plötzlich und unerwartet schneller Datenzugriff gefordert wird. Zusammenfassend ist festzuhalten: alle hier aufgezeigten Möglichkeiten sind immer noch besser, als die Datenauslagerung via Ausdrucken von csv – Extrakten. Auch diese Lösung haben wir in unserer Praxis bereits vorgefunden. Hier scheiterten wir jedoch an der Wirtschaftlichkeit der Altdatenmigration!

Kontaktadresse:
Mario Täuber
CSP GmbH & Co KG
Herrenäckerstraße 11
D-94431 Großköllnbach

Telefon:	+49 (0) 9953 3006 0
Fax:	+49 (0) 9953 3006 50
E-Mail	mario.taeuber@csp-sw.de
Internet:	www.datenbankarchivierung.de