

Partition-Exchange einiger abhängiger Entitäten - Konsistenzerhaltung im Fehlerfall
Dr. Kurt Franke
Cellent Finance Solutions GmbH
Stuttgart

Schlüsselworte

Datenbeladung, Massen-SQL, Historisierung, Partition-Exchange, DDL, Transaction, Autonomous Transaction, Konsistenz

Einleitung

Insbesondere wenn größere Datenmengen in historisierter Form vorgehalten werden müssen, ist es aus Performancegründen oft erforderlich, regelmäßige Datenanlieferungen, die auch selbst größere Mengen an Daten enthalten können, mit Massen-SQL-Statements in Arbeitstabellen aufzubereiten und anschließend via Partition-Exchange in die Partition für die aktuellen Daten der Zieltabelle zu integrieren. Gerade bei Historisierung auf Satzebene muss hier zuvor auch eine Zusammenführung mit den bereits zuvor dort vorhandenen Daten erfolgen. Dies lässt sich mit dem Oracle-Transactionshandling problemlos so durchführen, dass die Ergebnisse der gesamten Verarbeitung entweder komplett oder überhaupt nicht in die Zieltabelle integriert wurden. Dies ist hier möglich, weil ein via DDL verursachtes Commit einfach durch Einpacken in eine „Autonomous Transaction“ für außerhalb liegende DML vermieden werden kann und es bei einem Fehler in der Exchange-Partition-DDL keine durch DDL festgeschriebenen Daten gibt.

Komplexere Datenmodelle

Es bleibt jedoch selten bei einem solchen einfachen Fall. Oft gibt es Gruppierungen innerhalb einer Entität, die nach gutem relationalen Designstil in eigenen Tabellen zusammengefaßt werden und eine eigene Unterentität bilden. Als Beispiel sein hier eine Entität PARTNER (Geschäftspartner) aufgeführt, die sich im einfachsten Fall in NATUERLICHE_PERSON und JURISTISCHE_PERSON untergliedert. Auch eine weitere Entität mit Beziehungen zwischen den Partnern steht nicht unabhängig im Raum, sondern kann nur Beziehungen zwischen Partnern enthalten, die auch in PARTNER existieren.

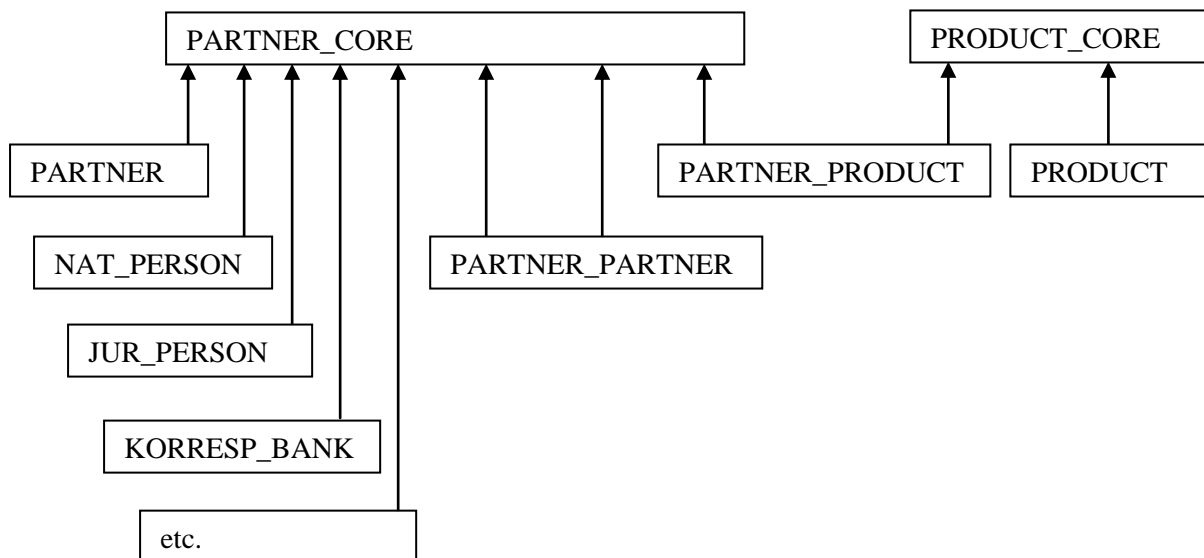


Abb. 1: Auszug aus einem etwas komplexeren Datenmodell – die Daten der Entitäten müssen zusammenpassen

Die CORE-Tabellen sind rein List-partitioniert nach GAE (Gefährdungs-Analyse-Einheit), die anderen Tabellen sind List-partitioniert nach GAE sowie Range-subpartitioniert nach GUELTIG_BIS (Ende Gültigkeitszeitraum eines Satzes aus der Historienabfolge). Alle für die Betrachtung hier relevanten Foreign Keys zeigen auf die CORE-Tabellen (natürlich haben alle diese Tabellen auch einen Foreign Key auf die GAE-Tabelle). Die Daten in dieser Struktur müssen den Daten aus dem Quell-System entsprechen und können nur dann als fachlich konsistent betrachtet werden, wenn dies erfüllt ist. Dies ist unabhängig davon, dass es auch Felder in den verschiedenen Entitäten gibt, für die die Datenhoheit im vorliegenden System angesiedelt ist.

Bei solchen zusammenhängenden Entitäten muss eine Verarbeitung sicherstellen, dass die Daten aus den verschiedenen Entitäten zueinander passen. Über Foreign Keys lässt sich absichern, dass in den abhängigen Entitäten – wieder am Beispiel von PARTNER betrachtet – keine Ausprägungen vorkommen, die nicht auch als Partner existieren. Dass alle real vorkommenden Ausprägungen der abhängigen Entitäten auch in den Daten abgebildet werden, muss programmtechnisch sichergestellt werden.

Auftreten einer neuartigen Problematik

In solchen Fällen mit mehreren voneinander abhängigen Entitäten ist einfaches Handling wie im ersten Absatz beschrieben nicht mehr möglich. Auch bei mehreren Partition Exchanges kann natürlich die Haupt-Transaction durch die Verwendung von „Autonomous Transaction“ vor den DDL-Commits geschützt werden. Das ist aber hier nicht ausreichend. Die Ergebnisse der einzelnen Partition Exchanges sind als DDL-Action nicht mehr zurückrollbar. Grundsätzlich kann nicht ausgeschlossen werden, dass die Integration der neu verarbeiteten Daten in die Zieltabellen für mehrere Entitäten erfolgreich durchgeführt wurde und dann bei einer weiteren Entität ein aktuell nicht handlebarer Fehler auftritt. Damit ist bereits ein Zustand eingetreten, bei dem eine bis mehrere Entitäten einen aktuelleren Stand aufweisen als eine bis mehrere anderen Entitäten und diese Daten damit nicht mehr zusammenpassen.

Da wegen der Verwendung von Exchange-Partition für den jeweils letzten Integrationsschritt in die Zieltabellen automatisch Commits wegen des DDL-Statements durchgeführt werden, ist das Oracle-Transactionshandling alleine hier nicht ausreichend. Das ist unabhängig davon, ob die Exchange-Partition-DDL-Actionen jeweils in eine „Autonomous Transaction“ verpackt sind oder nicht. Damit würde zwar verhindert, dass ein Commit für die außerhalb stattfindende DML erfolgt, aber dies ist eben nicht die gesamte Datenänderung. Ein wesentlicher Teil der Datenänderung erfolgt ja gerade durch die Integration in die Zieltabellen von Exchange-Partition-DDL, und der kann eben dadurch nicht vor einem DDL-Commit geschützt werden.

Ein mögliches Problemhandling

Es muss also in solchen Fällen ein übergreifender Mechanismus verwendet werden, der bei einem auftretenden nicht handlebaren Fehler den Zustand vor dem Beginn einer Verarbeitung wiederherzustellen. Am einfachsten ist es hier, für die Entitäten mit den erfolgreich integrierten Daten den Partition-Exchange rückgängig zu machen. Man muss dazu nur eine Möglichkeit zur Erkennung erfolgreich integrierter Daten einsetzen. Da die Befüllung der Exchange-Tables über ein einziges INSERT-Statement in der gleichen Verarbeitung wie der nachfolgende Partititon-Exchange erfolgt, haben alle diese Daten eine neuere SCN (System-Change-Number) als alle Daten aus der ursprünglichen Partition, die via Exchange Partition ausgetauscht wurde. Deshalb kann hier davon ausgegangen werden, dass jede mit ora_rowscn abgefragte SCN der neu befüllten Tabelle neuer ist als jede mit ora_rowscn abgefragte SCN aus der ursprünglichen Partition. Es genügt also in eine solchen Fall, genau eine beliebige Row aus der Exchange-Tabelle sowie aus der

ausgetauschten/auszutauschenden Partition abzufragen und die erhaltene SCN zu vergleichen, um festzustellen, ob ein Austausch bereits erfolgt ist oder nicht. Ist die aus der Partition erhaltene SCN größer als die aus der Tabelle, dann ist ein Austausch bereits erfolgt, andernfalls noch nicht. Ein Rückgängig-machen des Exchange Partition zur Wiederherstellung eines konsistenten Zustandes darf natürlich nur in den Fällen erfolgen, wenn ein Austausch bereits durchgeführt wurde. Da dieser elegante Oracle-Mechanismus hier genau die benötigte Funktionalität zur Überprüfung auf einen erfolgten Partition-Exchange in hoch-performer Form bereitstellt, ist eine Eigen-Implementierung hierfür nicht erforderlich.

Ein solches Zurücksetzen in den letzten konsistenten Zustand wird grundsätzlich sofort in jeder von einem Konsistenzproblem betroffenen Verarbeitung selbst durchgeführt. Erst wenn dieser Versuch fehlschlägt, muss die Konsistenz-Wiederherstellung nachfolgend über einen separaten Aufruf durchgeführt werden.

Handling von Session-Abbrüchen durch Fatal Error (ORA-00600 etc.)

Grundsätzlich ist es nicht ausgeschlossen, dass die verarbeitende Session bei einem Fehler mit einem Fatal-Error terminiert wird. Auch in einem solchen Falle muss sichergestellt sein, dass keinesfalls eine nachfolgende Verarbeitung für den betroffenen Teilbereich laufen kann, bevor die Ergebnisse aus der abgebrochenen Verarbeitung rückgängig gemacht werden konnten. Deshalb muss ein spezieller Statuswert bei Beginn der Partition-Exchange-Operation in einer Tabelle festgeschrieben hinterlegt werden, der bei erfolgreicher Beendigung oder erfolgreichem internen Undo der abgebrochenen Verarbeitung zur Wiederherstellung der Konsistenz zurückgesetzt wird. Bei einem Session-Abbruch bleibt dieser Status bestehen und stellt ein geeignetes Nachfolgehandling sicher. Da die Verarbeitungen für verschiedene GAE's komplett voneinander unabhängig sind, wird der Konsistenz-Status separat für jede GAE hinterlegt, damit die anderen GAE's nicht durch eine einzige inkonsistente GAE blockiert werden.

Die Überprüfung auf den Status KONSISTENT = 'N' ist in das Lock-Handling für die Verarbeitungen integriert – eine Allocation eines Locks wird abgelehnt, wenn ein solcher Status für den zu verarbeitenden Teilbereich existiert. Lediglich ein spezieller Lock für den Aufruf zur Wiederherstellung der Konsistenz ist dann allocierbar. Intern wird zur Konsistenz-Wiederherstellung die gleiche Utility verwendet, die auch schon in der Problem-Verarbeitung selbst zum Einsatz kommt. Neu ist hier lediglich der externe Aufruf und die Verwendung eines speziellen Locks, der es ermöglicht, diesen Aufruf dann – und nur dann – wenn ein Konsistenzproblem vorliegt, einzusetzen. Wenn die Wiederherstellung der Konsistenz erfolgreich war, wird der Status wieder auf KONSISTENT = 'J' zurückgesetzt.

Letztendlich wird sogar das Einspielen einer neuen Version der Software verhindert, die ja eventuell eine Änderung der Struktur von betroffenen Tabellen durchführen würde. Durch eine solche Änderung würde ein Partition-Exchange zurück zum ursprünglichen Zustand unmöglich gemacht. Um eine solche Modifizierung durch ein Update der Software bei Erhaltung der Funktionalität zu erlauben, müssten alle existierenden Exchange-Tables genauso wie die Original-Tabellen geändert werden. Dies würde die Update-Scripte deutlich komplexer gestalten, ohne einen echten Mehrwert an Funktionalität zu bieten. Da ist es einfacher, sicherzustellen, dass ein Update erst erfolgen kann, wenn die Konsistenz wiederhergestellt wurde.

Zusätzlicher Benefit des Verfahrens zur Konsistenz-Wiederherstellung

Das Rückgängig-machen der Partition-Exchange-Operation ist grundsätzlich nicht beschränkt auf eine Anwendung im Fehlerfalle. Sobald ein solches Verfahren implementiert ist, bietet sich auch die

Möglichkeit der Verwendung als Flashback-Verfahren zu einem Zustand vor der letzten erfolgreichen Verarbeitung an. Dazu muss nur noch eine geeignete Steuerung implementiert werden, während der Mechanismus selbst unverändert eingesetzt werden kann. Damit können dann fehlerhafte Datenlieferungen, solange sie noch im letzten Verarbeitungslauf integriert wurden, einfach und schnell wieder aus dem System entfernt werden.

Zusammenfassung

Partition-Exchange bietet ein hochperformantes Verfahren zur Integration neu bereitgestellter Daten in ein System. Sobald dabei jedoch mehrere voneinander abhängige Entitäten betroffen sind, kann es zu Problemen mit der Konsistenz dieser Entitäten untereinander kommen, wenn eine Exception beim Partition-Exchange einer Entität auftritt, nachdem die neuen Daten einer anderen Entität bereits erfolgreich ins System integriert wurden. Ein derartiges Problem kann wegen der nicht zurückrollbaren Datenintegration via DDL grundsätzlich nicht mit dem Oracle-Transactions-Mechanismus beherrscht werden. Deshalb ist ein speziell implementiertes Handling notwendig, das den letzten konsistenten Status wiederherstellt, indem es bereits erfolgte integrierte Daten durch Rückgängigmachen der Partition-Exchanges wieder aus dem System entfernt.

Kontaktadresse:

Dr. Kurt Franke
Cellent Finance Solutions GmbH
Calwer Straße 33
D-70173 Stuttgart

Telefon:	+49 (0) 711-222992-676
Fax:	+49 (0) 711-222992-899
E-Mail	Kurt.Franke@cellent-fs.de
Internet:	www.cellent-fs.de